

# A high-order numerical algorithm for DNS of low-Mach-number reactive flows with detailed chemistry and quasi-spectral accuracy

E. Motheau<sup>a,\*</sup>, J. Abraham<sup>a,b</sup>

<sup>a</sup> School of Mechanical Engineering, University of Adelaide, Adelaide, SA 5005, Australia

<sup>b</sup> School of Mechanical Engineering, Purdue University, 585 Purdue Mall, West Lafayette, IN 47907-2088, USA

## ARTICLE INFO

### Article history:

Received 18 June 2015

Received in revised form 19 February 2016

Accepted 24 February 2016

Available online 27 February 2016

### Keywords:

DNS

Low-Mach-number

Detailed chemistry

Turbulent reacting flow

High-order methods

Spectral accuracy

Operator splitting

## ABSTRACT

A novel and efficient algorithm is presented in this paper to deal with DNS of turbulent reacting flows under the low-Mach-number assumption, with detailed chemistry and a quasi-spectral accuracy. The temporal integration of the equations relies on an operating-split strategy, where chemical reactions are solved implicitly with a stiff solver and the convection–diffusion operators are solved with a Runge–Kutta–Chebyshev method. The spatial discretisation is performed with high-order compact schemes, and a FFT based constant-coefficient spectral solver is employed to solve a variable-coefficient Poisson equation. The numerical implementation takes advantage of the **2DECOMP&FFT** libraries developed by [1], which are based on a pencil decomposition method of the domain and are proven to be computationally very efficient. An enhanced pressure-correction method is proposed to speed up the achievement of machine precision accuracy. It is demonstrated that a second-order accuracy is reached in time, while the spatial accuracy ranges from fourth-order to sixth-order depending on the set of imposed boundary conditions. The software developed to implement the present algorithm is called **HOLOMAC**, and its numerical efficiency opens the way to deal with DNS of reacting flows to understand complex turbulent and chemical phenomena in flames.

© 2016 Elsevier Inc. All rights reserved.

## 1. Introduction

The rapid growth of computational capabilities in the last decades has allowed the application of high-fidelity numerical methods to unsteady reactive turbulent flows. For example, Large-Eddy Simulations (LES) are now commonly employed as a predictive tool in realistic complex configurations of interest to industry (see Pitsch [2], Gicquel et al. [3], Motheau et al. [4], among others). However, such methods rely on several models to take into account the unresolved physics. The development of turbulent combustion models that can be employed in practical applications requires high-fidelity data with which model results can be compared. In some cases, experimental results are available and very useful, but they often lack detailed information about the many species and temperature that are involved in chemical kinetics and about flow parameters like strain rate and turbulent statistics. Hence, Direct Numerical Simulation (DNS) is a powerful tool to

\* Corresponding author. Present address: Lawrence Berkeley National Laboratory, MS 50A-1148, 1 Cyclotron Rd., Berkeley, CA 94720, USA.

E-mail address: [emotheau@lbl.gov](mailto:emotheau@lbl.gov) (E. Motheau).

generate the data required to develop models. Note also that in many applications such as gas turbines and reciprocating engines, ignition and extinction are important limit phenomena that effect system performance. The accurate prediction of these limit phenomena requires detailed kinetics. Moreover, quasi-spectral accuracy is needed to generate the high-fidelity data.

In principle, whether LES or DNS approaches are employed, the most accurate representation of the physics would be achieved through the solution of the full set of compressible, reactive Navier–Stokes equations with detailed chemistry models. The computational cost of solving the compressible equations can be however prohibitive. One reason is the complexity of the chemistry. The reacting mixture generally includes a large number of chemical species with many hundreds of chemical reactions. The solution of the reaction equations leads to complex, stiff systems of equations that must be solved with dedicated numerical algorithms. In practice, simple models with reduced mechanisms are often employed to represent the chemistry with a very few number of species and chemical reactions (see Franzelli et al. [5] for an example). Detailed chemistry is generally limited to DNS of fundamental problems in canonical configurations (see Mukhopadhyay and Abraham [6] for an example).

Furthermore, from a numerical point of view, it is well known that the fully compressible approach presents several issues when dealing with flows featuring regions where the Mach number  $M$  is small, which is often the case in combustion process [7]. When  $M \rightarrow 0$  the equations tends to a singularity that break the stability and accuracy of numerical schemes employed to solve the set of fully compressible Navier–Stokes equations. As pointed out by Volpe [8], either with an explicit or an implicit approach for the time discretisation, when the Mach number is decreased to very small values the numerical errors increase as well as the convergence rate deteriorates. Furthermore, the critical time-step scales with the Mach number through a CFL condition so as to ensure the numerical stability due to the presence of acoustic waves. Hence, low-Mach-number regions in the flow will constrain the time-step to impracticable small values, which is a waste of time and computational resources in practice.

Several methods have been proposed to handle this particular issue of low-Mach-number regions in flows. Basically two approaches can be distinguished:

- The preconditioning methods, in which the time-derivatives in the Navier–Stokes equations are modified so as to optimise the eigenvalues of the compressible system by reducing the disparities between hydrodynamic and acoustic wave speeds. However, because the time-derivatives are modified, these methods are preferably applied to steady-state problems [9]. Otherwise, they can be embedded through dual time-stepping algorithms [10,11]. Most restricting, these methods cannot be generalised and must be properly designed for each particular problem investigated [12].
- The projection methods, where an equation for the pressure is solved to enforce a divergence constraint on the velocity field. Originally developed by Chorin [13] as a fractional-step method to solve incompressible flows, two different approaches were derived by reformulating the compressible Navier–Stokes equations:
  - under the low-Mach-number assumption [14,15], which removes the acoustic waves from the equations and only keeps hydrodynamic and entropic fluctuations;
  - with a scaling with respect to the pressure, leading to the so-called *Mach-uniform* or *all-speed* methods [16–18].

The domain of validity of the first approach is of course restricted to low-Mach-number flows and no acoustic phenomena can be captured, while the second approach aims to provide a more general method that can handle flows featuring a wide range of Mach numbers, from virtually zero to supersonic shockwaves, and where all compressibility effects are taken into account.

For DNS of combustion applications, the implementation of a detailed chemistry solver and high-order discretisation methods is not straightforward. Whereas the Mach-uniform method has received little attention [19,20] in the past few years, the low-Mach-number approach has received more attention [21–23]. Most algorithms published in the past are based on a fractional-step approach. Different ways to arrange the conservation equations are possible, but as recalled by Knikker [15] in his review paper, it is not possible to solve all of them in a conservative form unless an implicit approach is employed. This is, however, impractical in the context of DNS with detailed chemistry. For example Najm et al. [24], followed by Knio et al. [25], proposed to sacrifice the energy equation, solving it in its non-conservative form in the whole algorithm. Taking a different approach, Day and Bell [26] proposed a complex algorithm to solve the energy equation in its conservative form, but by compromising on the equation of state. This then required the implementation of a damping source term to control the deviation of the computed solution away from one which satisfies the equation of state. The developments that followed these seminal works were focused on the implementation of adaptative mesh refinement [27] or the design of more efficient and stable algorithms [28], especially for the treatment of the diffusion terms as well as the pressure equation appearing in the projection-correction procedure.

The aim of the present paper is to introduce an efficient algorithm, with optimised numerical methods, to perform high-fidelity DNS of reacting flows under the low-Mach-number assumption, with detailed chemistry and quasi-spectral accuracy. The first innovation of the present work is to compose a novel algorithm by selecting appropriate strategies and numerical methods from the prior works discussed above. The second innovation is that high-order discretisation schemes with spectral-like resolution are employed for all the variables solved. The pressure is solved with an FFT solver by means of the so-called spectral equivalence principle. To the authors' knowledge such accuracy for the pressure has not been reached before in the context of simulations of low-Mach-number combustion with detailed chemistry. Finally, the third

significant innovation of the present paper is to propose an enhanced efficient method to solve the variable-coefficient pressure equation by employing a constant-coefficient spectral solver, allowing speed-up in achieving machine precision accuracy.

The present paper is organised as follows. First, the governing reactive Navier–Stokes equations under the low-Mach-number assumption are presented in Sec. 2. Next, the particular issues related to the treatment of the detailed chemistry, the diffusion terms, and the solution of the pressure equation are presented in Sec. 3. A description of the whole algorithm and its implementation in the **HOLOMAC** (High-Order Low-Mach number Combustion) software is then given. In Sec. 4, the performance and accuracy of the algorithm are assessed with the help of several test problems. It is demonstrated that a second-order accuracy is achieved in time, while the spatial accuracy ranges from fourth-order to sixth-order, depending on the set of imposed boundary conditions.

## 2. Governing equations

According to Giovangigli [29], a dimensional analysis of the momentum equation that appears in the set of fully compressible reactive Navier–Stokes equations (see Poinso and Veynante [30] for further developments) reveals that the spatial gradient of the pressure  $p$  is of order  $\mathcal{O}(M^2)$  so that

$$p(\mathbf{x}, t) = p_0(t) + \underbrace{p_1(\mathbf{x}, t)}_{\mathcal{O}(M^2)}, \quad (1)$$

where  $p_0(t)$  is spatially uniform and  $p_1(\mathbf{x}, t)$  is the fluid dynamic perturbation, while  $M$  is the Mach number. Introducing the pressure split expressed in Eq. (1) in the state equation and taking the asymptotic limit  $M \rightarrow 0$  leads to the following simplified state law:

$$p_0 = \rho T \frac{\mathcal{R}}{\bar{W}}, \quad (2)$$

where  $\rho$  is the density,  $T$  the temperature,  $\mathcal{R}$  the universal gas constant and  $\bar{W}$  the mean molecular weight defined as

$$\bar{W} = 1 / \sum_{s=1}^{N_s} \frac{Y_s}{W_s}. \quad (3)$$

Here,  $W_s$  and  $Y_s$  are the molecular weight and the mass fraction of the species  $s$ , respectively,  $N_s$  being the total number of species present in the mixture.

The meaning of Eq. (2) is that the thermodynamic pressure  $p_0$  is constant in space (but may vary in time) and it is decoupled from the fluctuating part  $p_1$ . As the Mach number is considered small, acoustic fluctuations are neglected and  $p_1(\mathbf{x}, t)$  only embeds hydrodynamic perturbations. A rigorous mathematical derivation may be found in the seminal work of Majda and Sethian [21]. Conservation equations of continuity, momentum, energy and species transport can then be recast in the following low-Mach-number formulation, respectively:

$$\frac{\partial \rho}{\partial t} = - \frac{\partial \rho u_i}{\partial x_i}, \quad (4)$$

$$\frac{\partial \rho u_i}{\partial t} = C_{u_i} + D_{u_i} - \frac{\partial p_1}{\partial x_i}, \quad (5)$$

$$\frac{\partial T}{\partial t} = C_T + D_T + R_T + \frac{1}{\rho C_p} \frac{dp_0}{dt}, \quad (6)$$

$$\frac{\partial Y_s}{\partial t} = C_{Y_s} + D_{Y_s} + R_{Y_s} \quad s = 1, 2, \dots, N_s, \quad (7)$$

where  $u_i$  and  $x_i$  are the velocity and spatial coordinate along the  $i$ -th direction, respectively.

The convection  $C$  and diffusion  $D$  terms in Eqs. (5)–(7) are defined as

$$C_{u_i} = - \frac{\partial \rho u_j u_i}{\partial x_j}, \quad D_{u_i} = \frac{\partial}{\partial x_j} \left( \mu \left( \frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) - \frac{2}{3} \mu \frac{\partial u_k}{\partial x_k} \delta_{ij} \right), \quad (8)$$

$$C_T = -u_i \frac{\partial T}{\partial x_i}, \quad D_T = \frac{1}{\rho C_p} \left( \frac{\partial}{\partial x_i} \left( \lambda \frac{\partial T}{\partial x_i} \right) \right) + \frac{1}{\rho C_p} \left( \rho \sum_{s=1}^{N_s} C_{p,s} V_{s,i} Y_s \right) \frac{\partial T}{\partial x_i}, \quad (9)$$

$$C_{Y_s} = -u_i \frac{\partial Y_s}{\partial x_i}, \quad D_{Y_s} = \frac{1}{\rho} \frac{\partial \rho V_{s,i} Y_s}{\partial x_i}, \quad (10)$$

with

$$V_{s,i} Y_s = \mathcal{D}_s \frac{\partial Y_s}{\partial x_i} - Y_s \sum_{n=1}^{N_s} \mathcal{D}_n \frac{\partial Y_n}{\partial x_i}. \quad (11)$$

The source terms from chemical reactions are given by:

$$R_T = -\frac{1}{\rho C_p} \sum_{s=1}^{N_s} h_s \dot{\omega}_s, \quad R_{Y_s} = \frac{1}{\rho} \dot{\omega}_s. \quad (12)$$

In the above equations,  $\mu$  and  $\lambda$  are the dynamic viscosity and heat conductivity of the mixture, respectively, while  $C_{p,s}$  and  $C_p$  are the specific heat capacity at constant pressure for the species  $s$  and the total mixture, respectively, and are related by the following expression:

$$C_p = \sum_{s=1}^{N_s} Y_s C_{p,s}. \quad (13)$$

The terms  $\dot{\omega}_s$  and  $h_s$  represent the mass production rate and enthalpy, respectively, of the species  $s$ . The enthalpy term  $h_s$  is expressed as:

$$h_s = \int_{T_0}^T C_{p,s} dT + \Delta h_{f,s}^0, \quad (14)$$

where  $\Delta h_{f,s}^0$  is the enthalpy of formation of the species  $s$  at  $T_0 = 298.15$  K. Finally,  $\mathcal{D}_s$  is the mass diffusivity of the species  $s$ . Rigorous evaluation of  $\mathcal{D}_s$  is very expensive and several numerical strategies have been proposed to significantly reduce the computational burden (see Ern and Giovangigli [31], Magin and Degrez [32]). These methods have generally been applied to high-speed flows. In the present work, for ease of implementation, diffusion is approximated to first-order by using an effective-diffusivity multicomponent diffusion model formulated under the Hirschfelder–Curtiss [33] assumption:

$$\mathcal{D}_s = \frac{1 - Y_s}{\sum_{j \neq s} X_j / \mathcal{D}_{js}}. \quad (15)$$

Here  $\mathcal{D}_{js}$  is the binary diffusion coefficient between species  $j$  and  $s$ , and  $X_s$  is the mole fraction of species  $s$ . A well-known issue with this latter assumption is that the mass is not conserved, but the global mass can be maintained by adding a correction velocity term [30]. In the present mathematical formulation, this term appears in the right hand side of Eq. (11). Note also that Soret (molecular species diffusion due to temperature gradients) and Dufour (heat flux due to species mass fraction gradients) effects have been neglected.

The set of Eqs. (5)–(7) is subject to a constraint on the velocity field, which will be used in the pressure-projection part of the fractional-step algorithm. This constraint can be derived by reformulating the continuity equation (4) as

$$\frac{\partial u_i}{\partial x_i} = -\frac{1}{\rho} \frac{D\rho}{Dt}, \quad (16)$$

where the RHS of Eq. (16) is obtained by differentiating the equation of state along particle paths, leading to:

$$\frac{\partial u_i}{\partial x_i} = \frac{1}{T} \frac{DT}{Dt} + \sum_{s=1}^{N_s} \frac{\bar{W}}{W_s} \frac{DY_s}{Dt} - \frac{1}{p_0} \frac{dp_0}{dt}. \quad (17)$$

Replacing the material derivatives that appear in Eq. (17) by their expressions in Eqs. (6) and (7) leads to the following expression of the velocity constraint:

$$\frac{\partial u_i}{\partial x_i} = \frac{dp_0}{dt} \tilde{P} + \tilde{D}, \quad (18)$$

where

$$\tilde{P} = \frac{1}{\rho C_p T} - \frac{1}{p_0}, \quad (19)$$

$$\tilde{D} = \frac{1}{T} (D_T + R_T) + \sum_{s=1}^{N_s} \frac{\bar{W}}{W_s} (D_{Y_s} + R_{Y_s}). \quad (20)$$

In the context of a simulation with open boundaries, typically inflow/outflow conditions, the thermodynamic pressure  $p_0$  is static in time and set by atmospheric conditions. Consequently, the term  $dp_0/dt$  vanishes in Eq. (18). However if the computational domain is closed, the thermodynamic pressure  $p_0$  may change in time. As the total mass remains constant

through the domain and is equal to the volume integral of the density,  $p_0$  can be expressed with the help of Eq. (2) and reads:

$$p_0 = \frac{M_0 \mathcal{R}}{\int_V \left( T \sum_{s=1}^{N_s} \frac{Y_s}{W_s} \right)^{-1} dV}, \quad (21)$$

with  $M_0 = \int_V \rho \, dV$ . As pointed out by Nicoud [14], the time derivative of  $p_0$  may be expressed by integrating Eq. (18) over a domain  $V$  to give the following equation:

$$\frac{dp_0}{dt} = \left( \int_V \partial u_i / \partial x_i \, dV - \int_V \tilde{\mathcal{D}} \, dV \right) / \int_V \tilde{\mathcal{P}} \, dV. \quad (22)$$

Since  $\int_V \partial u_i / \partial x_i \, dV = \int_S u_i^n dS$ ,  $S$  being the surface boundary, the velocities along normal  $n$  sum up to zero if hard walls or periodic boundary conditions are imposed everywhere. Hence, the term  $\int_V \partial u_i / \partial x_i \, dV$  in Eq. (22) vanishes.

### 3. Numerical methods

#### 3.1. Overall presentation of the algorithm

The algorithm developed in the present paper is based on a fractional-step, segregated method. Basically the procedure can be summarised by the following two parts:

1. The thermochemical system composed of the energy Eq. (6) and the species transport Eq. (7) is solved first. During this step, temperature and species mass fractions are advanced in time, making it possible to compute a new density through the equation of state (2).
2. With this new density, the momentum Eq. (5) can be integrated to advanced in time the velocity fields. This step relies on a projection/correction method, where a Poisson equation for the pressure is solved to enforce the divergence condition imposed by the continuity equation (16).

##### 3.1.1. Operator-split techniques, stiff integration and treatment of the diffusion

A particular issue encountered in Part 1 is with the wide range of time scales involved in the different operators comprising the thermochemical system. Indeed, due to the detailed chemistry, the evaluation of the chemical reactions is very sensitive to the state variables and it forms a stiff system of ODEs to solve, which requires dedicated numerical methods that belong to the class of *stiff solver* (see Hairer and Wanner [34] for a review). However, the set of ODEs has convection and diffusion operators, and including them into a stiff integrator would be particularly inefficient. The popular strategy employed to cope with this numerical challenge is to solve each term separately through an operator-split scheme. Knio et al. [25] and Day and Bell [26] implemented the so-called *Strang operator-split* (see Strang [35]), which is second order in time, and demonstrated its efficiency to solve the thermochemical system while keeping large time-steps and an acceptable computational cost. As recalled by Duarte et al. [36], a particular attraction of this strategy is that each operator can be solved with its own specific numerical method, opening the way to tailor an overall algorithm.

In the algorithm developed in the present paper, the following operator-split scheme is retained:

$$\mathbf{H}(t^n + \Delta t^n) = \mathcal{H}_{dt/2}^{C-D} \mathcal{H}_{dt}^R \mathcal{H}_{dt/2}^{C-D} \mathbf{H}(t^n), \quad (23)$$

where  $\mathbf{H}$  is the solution vector while  $\mathcal{H}$  refers to the operators of convection (C), diffusion (D) and reaction (R). Another combination  $\mathcal{H}_{dt/2}^R \mathcal{H}_{dt}^{C-D} \mathcal{H}_{dt/2}^R$  was tested and compared to Eq. (23) for the freely propagating methane/air flames cases presented in §4.1, but virtually no difference was noticed in the solutions. It is emphasised that within the order of time-steps employed for the whole algorithm, errors occurring in the operator-splitting procedure are negligible. Consequently, the operator-split scheme described at Eq. (23) is selected because as the evaluation of the reaction operators requires a significant computational effort, it is practically more time-efficient to solve it only once.

A second issue arising in Part 1 is the choice of a numerical method for the integration of the diffusion operators. It is well known that the evaluation of diffusion imposes a limit on the time-step to ensure the stability of the algorithm. This limitation can be reduced by treating the diffusion terms with an implicit method. However, as recalled by Najm and Knio [37], the diffusion terms exhibit a non-linear dependence on the temperature and species mass fractions, and are strongly coupled to the reaction operators. Hence, relying on an implicit method is particularly unattractive within the context of solving detailed reacting flows. To overcome this issue, Day and Bell [26] proposed to solve a temperature equation prior to an enthalpy equation in order to provide fluid properties at the new time level. Another possibility proposed by Knio et al. [25] is to solve explicitly the diffusion operators by decomposing the time integration in smaller fractional steps. However the major drawback of this approach is the large number of iterations required to ensure stability. For example, Yu et al. [28] reported the need to use more than one hundred fractional steps for the simulation of a hydrogen/air flame. Such poor performance motivated Najm and Knio [37] to adopt an explicit Runge–Kutta–Chebyshev (RKC) method [38] instead of their

initial choice based on the fractional-steps approach. They reported a number of iterations of approximately 32, which is a significant gain of computational time. However in this latter work, the authors reformulated the temperature equation into an evolution equation for the density, requiring evaluation of additional operators, which is not computationally efficient (see the discussion in the review of Knikker [15]). In the present paper, the temperature and species mass fraction equations are kept in their original form, and the RKC method is selected to integrate the  $\mathcal{H}^{C-D}$  operators. Results presented in Sec. 4 show that only a dozen (approximately) iterations for the RKC method are enough to ensure stability, even for relatively high CFL numbers of approximately 0.85. This value is close to the stability limit for explicit time integration. Moreover, for all cases studied, an estimation based on the von Neumann stability criterion for the diffusion has been conducted. It shows that, compared to the RKC method, the maximum critical time-step would be about four orders of magnitude smaller with a forward explicit time-integration. Thus, the results demonstrate that the RKC method is very efficient to deal with relatively larger time-steps. Note that the fluid properties and chemical reactions are computed with the **CHEMKIN** libraries [39]. As pointed out by Najm and Knio [37], the repeated evaluation of fluid properties through these libraries is computationally costly and they proposed to rely on extrapolation/interpolation techniques instead. Such an approach, particularly efficient and introducing negligible errors, is adopted in the present paper.

Particular attention must be drawn to the convection operators. Indeed, they all involve the velocity vector, which is however not known at the next time-step. Thus, repeated evaluations of the convection operators at each RKC stage would require an extrapolation of the velocities. Instead, the time integration of the convection operators is performed once with a linear multistep method, namely a second-order Adams–Bashforth (AB2) scheme. Hence, convection terms are imposed during the integration of diffusion operators as constant source terms. Yu et al. [28] pointed out that merging the term appearing in the right hand side of  $D_T$  (see Eq. (9)) into the diffusion integration procedure would increase the computational burden. Instead, they merged this term with the convection operator to impose it as a constant source term. It was found in the development of the present algorithm that such a choice leads locally to a lack of conservation of the mass, reducing strongly the accuracy of the algorithm. Hence, despite the additional computational cost, this term is repeatedly evaluated during the integration procedure to ensure mass conservation.

### 3.1.2. Projection methods and treatment of the pressure

The pressure-projection step employed in Part 2 is widely pointed out in the literature as posing a particular difficulty, playing a crucial role in determining the numerical stability [14,15]. The origin of this method goes back to the fractional-step, projection method developed for incompressible flows by Chorin [13]. Basically, the velocity and the pressure are decoupled in the momentum equation (5); after the time advancement of the velocity, a Poisson equation for the pressure is solved to project the intermediate velocity onto a space that enforces a divergence-free constraint. In the context of low-Mach-number flows, the variable density appears in the Poisson equation. In the fractional-step method, the constraint on the velocity field is imposed from already known values computed at the new time level, and can be formulated in two ways:

- A first approach is to impose the divergence of the momentum ( $\rho u$ ), which is known through the continuity equation (4). This amounts to computing and imposing the time derivative of the density at the new time level. It leads to a *constant-coefficient* Poisson equation requiring the evaluation of the operator  $\nabla^2$ .
- A second approach is to impose the divergence of the velocity  $u$ , which is known by taking the material derivative of the equation of state, as detailed in Eqs. (16)–(17). It leads to a *variable-coefficient* Poisson equation requiring the evaluation of the operator  $\nabla(1/\rho\nabla)$ , where  $\rho$  varies spatially.

The numerical solution of the *constant-coefficient* Poisson equation is the most straightforward and can be done by means of direct solvers. However this approach presents two major drawbacks: first, it has been demonstrated by Nicoud [14] that in the inviscid limit, unlike the *variable-coefficient* Poisson equation approach, this formulation does not recover the divergence-free velocity constraint. Second, it has been widely reported in the literature that the time-derivative of density is a source of instability in the Poisson equation, limiting the method to small amplitude variations of the density (the ratio value of 3 is commonly reported). Najm et al. [24] proposed a predictor–corrector scheme that extended the stability to density ratios of 10. During the development of the present paper, this predictor–corrector approach has been tested and it has been observed that the stability also depends on the spatial steepness and time evolution of the density ratio. In the context of autoignition of an heptane/air mixture presented in §4.3, this approach has been found to be always unstable for density ratios greater than approximately 7.

Such disappointing results have motivated the adoption of an approach based on the *variable-coefficient* Poisson equation. Although not shown in the present paper, numerical tests have validated the strongly stable behaviour of this approach for large and steep density ratios. However, the major drawback reported in the literature is that this equation is challenging to solve numerically. In the context of low-Mach-number reactive flows, many of the algorithms published rely on iterative solvers. Yu et al. [28] and Safta et al. [27] used multigrid methods with a finite-difference discretisation, reaching second- and fourth-order accuracy, respectively. On the other hand, Knikker [15] employed a second-order Krylov-based solver but reached fourth-order accuracy by using an iterative residual correction method, the residuals being computed with compact schemes. Moreover, Desjardins et al. [40] reached similar high-order accuracy with a combination of spectral and Krylov-based methods. Unlike the previous works cited above, a constant-coefficient spectral solver based on fast Fourier transforms



and pencil domain decomposition is employed in this work as a novel contribution to the literature. It is shown in Sec. 4 that sixth-order accuracy is reached for the pressure.

The numerical methods presented in this paper are implemented in the **HOLOMAC** software, which is based on the **Incompact3D** framework [41,42] and the **2DECOMP&FFT** libraries [1]. Details about the spectral solver, its implementation and the parallelisation techniques can be found in the aforementioned references. Note that the **FFTW3** libraries [43] have been implemented to perform the Fast Fourier Transforms (FFT) employed by the spectral solver for the pressure. The mathematical description of the algorithm for the time integration, as well as the numerical methods for the spatial discretisation, are now presented below.

### 3.2. Temporal integration

#### 3.2.1. Step 1: computation of explicit terms

The first step of the present algorithm is to compute the explicit terms that will be imposed as source terms during the time integration of the diffusion operators with the RKC method. These source terms embed the explicit time integration of the convection terms  $C_{u_i}$ ,  $C_T$  and  $C_{Y_s}$  from the time-step  $n$  to  $n+1$  with an AB2 scheme, as well as the spatial gradient of hydrodynamic pressure  $p_1$  and the time derivative of the thermodynamic pressure  $p_0$  that appear in the momentum equation (5) and the energy equation (6), respectively. The source terms  $S_{u_i}$ ,  $S_T$  and  $S_{Y_s}$  for the momentum, energy and species transport equations are expressed as follows:

$$S_{u_i} = \left[ (1 + \zeta_t) C_{u_i}^n - \zeta_t C_{u_i}^{n-1} \right] - \frac{\partial p_1^n}{\partial x_i}, \quad (24)$$

$$S_T = \left[ (1 + \zeta_t) C_T^n - \zeta_t C_T^{n-1} \right] + \frac{1}{(1 + \zeta_t) \rho^n C_p^n - \zeta_t \rho^{n-1} C_p^{n-1}} \frac{\widetilde{dp}_0}{dt}, \quad (25)$$

$$S_{Y_s} = (1 + \zeta_t) C_{Y_s}^n - \zeta_t C_{Y_s}^{n-1}, \quad (26)$$

where

$$\frac{\widetilde{dp}_0}{dt} = \frac{\Delta t^{n-2} \left( 1 + \frac{\Delta t^n}{\Delta t^{n-1}} \right) p_0^n - \left( \Delta t^{n-2} + \Delta t^n + \frac{\Delta t^n \Delta t^{n-2}}{\Delta t^{n-1}} \right) p_0^{n-1} + \Delta t^n p_0^{n-2}}{\Delta t^{n-1} \Delta t^{n-2}} \quad (27)$$

is the extrapolation of the time derivative of the thermodynamic pressure computed over the new time-step, and with

$$\zeta_t = \frac{1}{2} \frac{\Delta t^n}{\Delta t^{n-1}}, \quad (28)$$

$$\Delta t^n = t^{n+1} - t^n, \quad (29)$$

$$\Delta t^{n-1} = t^n - t^{n-1}, \quad (30)$$

$$\Delta t^{n-2} = t^{n-1} - t^{n-2}. \quad (31)$$

Note that the evaluation of the time derivative of  $p_0$  through Eq. (22) would require all the diffusion and reaction operators to be computed twice, as well as two volume integrations over the full domain to be performed. In order to save computational time, this procedure is only performed at the first iteration of a simulation, the remaining one being computed with the discrete formulation expressed in Eq. (27). Of course in the case of an open domain with inflow/outflow boundary conditions,  $\frac{dp_0}{dt} = 0$ .

#### 3.2.2. Step 2: RKC integration of scalars diffusion terms over the first half time-step

The diffusion terms  $D_{u_i}$ ,  $D_T$  and  $D_{Y_s}$  are integrated from the time-step  $n$  to  $n+1/2$  with the explicit Runge–Kutta–Chebyshev (RKC) method presented by Verwer et al. [38]. The source terms computed in §3.2.1 are imposed during each iteration of the RKC method. The total number of iterations  $K$  is a free parameter chosen by the user, but the RKC scheme requires at least two iterations, viz.  $K \geq 2$ .

The starting values, denoted by superscript 0, are values at time-step  $n$ :

$$T^0 = T^n, \quad (32)$$

$$Y_s^0 = Y_s^n, \quad (33)$$

$$\rho^0 = \rho^n. \quad (34)$$

During the first stage of integration, denoted by superscript 1, values are updated using the following relations:

$$T^1 = T^0 + \tilde{\eta}_1 \frac{\Delta t^n}{2} \left( D_T^0 + S_T \right), \quad (35)$$

$$Y_s^1 = Y_s^0 + \tilde{\eta}_1 \frac{\Delta t^n}{2} \left( D_{Y_s}^0 + S_{Y_s} \right), \quad (36)$$

$$\rho^1 = \frac{p_0^n}{\mathcal{R}T^1 \sum_{s=1}^{N_s} \frac{Y_s^1}{W_s^1}}. \quad (37)$$

Integration over the remaining stages of integration, denoted  $k$  with  $k = 2, \dots, K$ , is performed by employing the following relations:

$$T^k = (1 - \eta_k - \nu_k) T^0 + \eta_k T^{k-1} + \nu_k T^{k-2} + \tilde{\eta}_k \frac{\Delta t^n}{2} \left( D_T^{k-1} + S_T \right) + \tilde{\gamma}_k \frac{\Delta t^n}{2} \left( D_T^0 + S_T \right), \quad (38)$$

$$Y_s^k = (1 - \eta_k - \nu_k) Y_s^0 + \eta_k Y_s^{k-1} + \nu_k Y_s^{k-2} + \tilde{\eta}_k \frac{\Delta t^n}{2} \left( D_{Y_s}^{k-1} + S_{Y_s} \right) + \tilde{\gamma}_k \frac{\Delta t^n}{2} \left( D_{Y_s}^0 + S_{Y_s} \right), \quad (39)$$

$$\rho^k = \frac{p_0^n}{\mathcal{R}T^k \sum_{s=1}^{N_s} \frac{Y_s^k}{W_s^k}}. \quad (40)$$

At the end of the integration, values at time-step  $n + 1/2$  are given by

$$T^{n+1/2} = T^K, \quad (41)$$

$$Y_s^{n+1/2} = Y_s^K, \quad (42)$$

$$\rho^{n+1/2} = \rho^K. \quad (43)$$

Note that in order to save a significant amount of computational time, the thermodynamic parameters  $\mathcal{D}_s$ ,  $C_p$  and  $\lambda$  are not computed at each stage through the **CHEMKIN** routines, but extrapolated from previous values (see §3.4 for more details).

The coefficients appearing in all stages of integration are given by

$$\tilde{\eta}_1 = b_1 \omega_1, \quad (44)$$

$$\eta_k = \frac{2b_k \omega_0}{b_{k-1}}, \quad \nu_k = \frac{-b_k}{b_{k-2}}, \quad \tilde{\eta}_k = \frac{2b_k \omega_1}{b_{k-1}}, \quad \tilde{\gamma}_k = -a_{k-1} \tilde{\eta}_k, \quad (45)$$

where  $a_k$ ,  $b_k$ ,  $\omega_0$  and  $\omega_1$  are given by

$$a_k = 1 - b_k \mathcal{T}_k(\omega_0), \quad (46)$$

$$b_0 = b_2, \quad b_1 = \frac{1}{\omega_0}, \quad b_k = \frac{\mathcal{T}_k''(\omega_0)}{(\mathcal{T}_k'(\omega_0))^2}, \quad (47)$$

with

$$\omega_0 = 1 + \frac{\epsilon}{K^2}, \quad \omega_1 = \frac{\mathcal{T}_k'(\omega_0)}{\mathcal{T}_k''(\omega_0)}. \quad (48)$$

Note that in the remainder of the present paper, the value of  $\epsilon = 10$  is chosen (see discussion in §4.4). Moreover,  $\mathcal{T}_l(x)$  is the first kind Chebyshev polynomials:

$$\mathcal{T}_k(x) = 2x\mathcal{T}_{k-1}(x) - \mathcal{T}_{k-2}(x), \quad (49)$$

$$\mathcal{T}_0(x) = 1, \quad (50)$$

$$\mathcal{T}_1(x) = x. \quad (51)$$

### 3.2.3. Step 3: stiff integration of reactive terms over a full time-step

During this step, the reactive terms  $R_T$  and  $R_{Y_s}$  are integrated over a full time-step  $\Delta t^n$ . These terms involve the computation of chemical reaction rates that are very sensitive to temperature and species mass fractions. Thus, it forms the following stiff system of ODEs:

$$\begin{aligned} T^{**} - T^* &= \int_{t^n}^{t^{n+1}} R_T dt, \\ Y_s^{**} - Y_s^* &= \int_{t^n}^{t^{n+1}} R_{Y_s} dt. \end{aligned} \quad (52)$$



The starting values are denoted by the superscript \* and are the ones computed at the end of §3.2.2. Note that as the chemical reactions do not involve a variation of mass, the density is kept constant during the whole integration stage, and is only updated at the end of the process, denoted by the superscript \*\*:

$$\rho^{**} = \frac{p_0^n}{\mathcal{R} T^{**} \sum_{s=1}^{N_s} \frac{Y_s^{**}}{W_s^{**}}} \quad (53)$$

Many stiff solvers are available in the literature to integrate Eq. (52). DVODE [44], RADAU5 [34] and SEULEX [34] have been implemented in the framework of **HOLOMAC**. All of these solvers rely on relative and absolute tolerance values to reach convergence. Assessment of performance and accuracy of these solvers is out of the scope of the present paper, and DVODE is retained for the remainder of the present paper, with both relative and absolute tolerance values set to  $10^{-14}$ , respectively. With such low values, it has been observed that all solvers give virtually the same results.

### 3.2.4. Step 4: RKC integration of scalars diffusion terms over the last half time-step

This step is very similar to §3.2.2, using  $T^{**}$ ,  $Y_s^{**}$  and  $\rho^{**}$  as initial values and the same number of iterations  $K$ . At the end of this step,  $T$  and  $Y_s$  are available at the new time  $t^{n+1}$  and boundary conditions for temperature and species mass fractions can be applied. Note that in the case of an outflow boundary condition, an Orlanski-type boundary condition [45] is employed to convect the physical variables out of the domain.

In the case of a closed domain, the new thermodynamic pressure  $p_0^{n+1}$  can be evaluated by applying Eq. (21). Hence, by knowing  $T^{n+1}$ ,  $Y_s^{n+1}$  and  $p_0^{n+1}$ , the new density  $\rho^{n+1}$ , diffusion coefficients  $\mathcal{D}_s^{n+1}$ ,  $\lambda^{n+1}$  and  $\mu^{n+1}$ , as well as  $C_p^{n+1}$  can now be updated via the **CHEMKIN** routines.

### 3.2.5. Step 5: computation of the divergence velocity constraint

Prior to the computation of the momentum equation, the divergence velocity constraint that will be applied during the pressure projection step can be computed with the final values at the end of §3.2.4. Similarly to §3.2.1, this velocity constraint could be discretised directly by applying Eqs. (18) and (22), but in order to save computational time and avoid redundant evaluation of diffusion and reaction operators, the semi-discrete approach proposed by Yu et al. [28] is adopted:

$$\begin{aligned} \frac{\partial u_i^{n+1}}{\partial x_i} = & -\frac{1}{p_0^{n+1}} \left. \frac{dp_0}{dt} \right|^{n+1} + \frac{1}{T^{n+1}} \left( (1 + \zeta_t^*) \frac{DT}{Dt}^{n+\frac{1}{2}} - \zeta_t^* \frac{DT}{Dt}^{n-\frac{1}{2}} \right) \\ & + \sum_{s=1}^{N_s} \frac{\bar{W}^{n+1}}{W_s} \left( (1 + \zeta_t^*) \frac{DY_s}{Dt}^{n+\frac{1}{2}} - \zeta_t^* \frac{DY_s}{Dt}^{n-\frac{1}{2}} \right), \end{aligned} \quad (54)$$

with

$$\left. \frac{dp_0}{dt} \right|^{n+1} = \frac{[(\Delta t^n + \Delta t^{n-1})^2 - (\Delta t^n)^2] p_0^{n+1} - (\Delta t^n + \Delta t^{n-1})^2 p_0^n + (\Delta t^n)^2 p_0^{n-1}}{\Delta t^n \Delta t^{n-1} (\Delta t^n + \Delta t^{n-1})}, \quad (55)$$

$$\frac{DT}{Dt}^{n+\frac{1}{2}} = \frac{T^{n+1} - T^n}{\Delta t^n} - \left[ (1 + \zeta_t) C_T^n - \zeta_t C_T^{n-1} \right], \quad (56)$$

$$\frac{DY_s}{Dt}^{n+\frac{1}{2}} = \frac{Y_s^{n+1} - Y_s^n}{\Delta t^n} - \left[ (1 + \zeta_t) C_{Y_s}^n - \zeta_t C_{Y_s}^{n-1} \right]. \quad (57)$$

Note that unlike the formulation of Yu et al. [28], Eqs. (54)–(57) include the possibility of time-step variations. In the above expressions,  $\zeta_t$  is defined by Eq. (28) and  $\zeta_t^*$  is expressed as:

$$\zeta_t^* = \frac{\Delta t^n}{\Delta t^n + \Delta t^{n-1}}. \quad (58)$$

### 3.2.6. Step 6: RKC integration of momentum diffusion terms over a full time-step

During this step, the momentum equation is advanced in time through the RKC integration of diffusion terms over a full time-step. Note that the source terms include the gradients of hydrodynamic pressure from the previous time-step. According to Guermond et al. [46], this technique leads to a correction in pressure of the velocity fields and increases the accuracy of the projection method to second-order. Note also that the stages of integration are denoted  $l = 2, \dots, L$ . The following relations are then employed:

$$(\rho u_i)^0 = (\rho u_i)^n, \quad (59)$$

$$(\rho u_i)^1 = (\rho u_i)^0 + \tilde{\eta}_1 \Delta t^n (D_{u_i}^0 + S_{u_i}), \quad (60)$$

$$(\rho u_i)^l = (1 - \eta_l - \nu_l) (\rho u_i)^0 + \eta_l (\rho u_i)^{l-1} + \nu_l (\rho u_i)^{l-2} + \tilde{\eta}_l \Delta t^n (D_{u_i}^{l-1} + S_{u_i}) + \tilde{\gamma}_l \Delta t^n (D_{u_i}^0 + S_{u_i}). \quad (61)$$

Similarly to the RKC integration of diffusion terms in the scalars equations described in §3.2.2, the density  $\rho$  and viscosity  $\mu$  are interpolated from  $t^n$  and newly known  $t^{n+1}$  values (see §3.4 for more details).

At the end of this step, the velocity field does not satisfy the divergence constraint formulated in Eq. (16), and is at a provisional state denoted by the superscript  $*$ :

$$(\rho u_i)^* = (\rho u_i)^L. \quad (62)$$

The boundary conditions for momentum are applied at the end of this step. In the case of an outflow boundary condition, an Orlanski-type boundary condition [45] is employed to convect the momentum  $(\rho u_i)$  out of the domain. Note that according to Gresho [47] the conservation of mass is enforced to machine precision accuracy by correcting the mass flux leaving the domain to exactly match the entering mass flux imposed at the inlet boundary condition.

### 3.2.7. Step 7: pressure projection

In the fractional-step method, the final velocity field  $u^{n+1}$  is obtained by correcting the provisional velocity field  $u_i^*$  with the gradients of the hydrodynamic pressure, which is obtained by solving the following variable-coefficient Poisson equation:

$$\frac{\partial}{\partial x_i} \left( \frac{1}{\rho^{n+1}} \frac{\partial p'}{\partial x_i} \right) = \frac{1}{\Delta t^n} \left( \frac{\partial u_i^*}{\partial x_i} - \frac{\partial u_i^{n+1}}{\partial x_i} \right), \quad (63)$$

where  $p' = p_1^{n+1} - p_1^n$ . Note that the last term of the RHS has been computed in §3.2.5 using Eq. (54).

In the present paper, three methodologies based on a FFT spectral solver are investigated to efficiently solve the non-linear Poisson equation (63):

- **Method I: fully implicit – iterative**

The basic brute-force approach suggested by Nicoud [48] is to solve Eq. (63) with the following iterative procedure:

$$\frac{\partial^2 p'_q}{\partial x_i^2} = \frac{\partial}{\partial x_i} \left[ \left( 1 - \frac{\rho_0^{n+1}}{\rho^{n+1}} \right) \frac{\partial p'_{q-1}}{\partial x_i} \right] + \frac{\rho_0^{n+1}}{\Delta t^n} \left( \frac{\partial u_i^*}{\partial x_i} - \frac{\partial u_i^{n+1}}{\partial x_i} \right). \quad (64)$$

Nicoud [48] recommends for the value of  $\rho_0^{n+1}$  a plane averaging of the density. However in the present algorithm, it was found that such a choice prevents the convergence of Eq. (64), which only converge for  $\rho_0^{n+1} = \min(\rho^{n+1})$ . Each sub-iteration  $q$  in Eq. (64) is solved exactly using a spectral solver based on Fast Fourier Transforms. Eq. (64) is considered converged when

$$\|p'_q - p'_{q-1}\| \leq \xi, \quad (65)$$

where  $\|\cdot\|$  is the  $\mathcal{L}^2$ -norm, and  $\xi$  is a tolerance parameter set by the user.

The provisional velocity field is then corrected by applying the following relation:

$$u_i^{n+1} = u_i^* - \frac{\Delta t^n}{\rho^{n+1}} \frac{\partial p'_q}{\partial x_i}, \quad (66)$$

where  $p'_q$  is the solution of the last iteration of Eq. (64).

As shown in §4.3, inaccuracy in the evaluation of the pressure leads to errors in the velocity field when applying Eq. (66). Resolution of the projection/correction step up to the machine precision then requires very low values for  $\xi$ , typically of the order of  $10^{-12}$ , leading to a significant total number of iterations  $q$ , typically of the order of several hundreds. Note that in order to accelerate the convergence, an initial starting value  $p'_0$  is provided with the help of the extrapolation  $p^{n+1} = 2p^n - p^{n-1}$ , so that:

$$p'_0 = (2p^n - p^{n-1}) - p^n. \quad (67)$$

- **Method II: semi-implicit – direct**

In order to save computational time, an interesting technique proposed by Dodd and Ferrante [49] is to split Eq. (63) into a variable part  $p'_0$  provided explicitly by an extrapolation of the pressure (see Eq. (67)), while the other term  $p'$  is solved within a constant-coefficient Poisson equation requiring only one evaluation of the FFT spectral solver, i.e.

$$\frac{\partial^2 p'}{\partial x_i^2} = \frac{\partial}{\partial x_i} \left[ \left( 1 - \frac{\rho_0^{n+1}}{\rho^{n+1}} \right) \frac{\partial p'_0}{\partial x_i} \right] + \frac{\rho_0^{n+1}}{\Delta t^n} \left( \frac{\partial u_i^*}{\partial x_i} - \frac{\partial u_i^{n+1}}{\partial x_i} \right). \quad (68)$$

Of course Eq. (68) is very similar to Eq. (64) in method I. The difference in method II is that the correction of the provisional velocity field now includes the contributions of both the implicit and explicit pressure terms, i.e.

$$u_i^{n+1} = u_i^* - \Delta t^n \left[ \frac{1}{\rho_0^{n+1}} \frac{\partial p'}{\partial x_i} + \left( \frac{1}{\rho^{n+1}} - \frac{1}{\rho_0^{n+1}} \right) \frac{\partial p'_0}{\partial x_i} \right]. \quad (69)$$

As demonstrated by Dodd and Ferrante [49], this method is efficient when the gradients of pressure are smooth in time, which is not always the case in combustion applications. As shown in §4.3 with a test case of autoignition of a heptane/air mixture, this method, while the fastest, leads to errors of order  $10^{-3}$  during the pressure correction step.

• **Method III: semi-implicit – iterative**

Method III is a mix of methods I and II. The pressure  $p'$  is computed with Eq. (64), while the correction of the velocity field is obtained with the following variant of Eq. (69):

$$u_i^{n+1} = u_i^* - \Delta t^n \left[ \frac{1}{\rho_0^{n+1}} \frac{\partial p'_q}{\partial x_i} + \left( \frac{1}{\rho^{n+1}} - \frac{1}{\rho_0^{n+1}} \right) \frac{\partial p'_{q-1}}{\partial x_i} \right]. \quad (70)$$

As shown in §4.3, this method is very efficient. Based on the two test cases investigated, the machine precision accuracy in the projection/correction step is achieved for values of the tolerance parameter  $\xi$  ranging from  $10^{-8}$  to  $10^{-6}$ , which represents a significant gain in the required total number of iterations and thus, computational time.

Once the final velocity field  $u_i^{n+1}$  is updated, the new hydrodynamic pressure  $p_1^{n+1}$  can be updated with the following relation:

$$p_1^{n+1} = p_1^n + p'_q. \quad (71)$$

At this step, the velocities  $u_i^{n+1}$ , temperature  $T^{n+1}$  and species mass fractions  $Y_s^{n+1}$  are spatially filtered (see §3.3) in order to remove small oscillations that could destabilise the algorithm. Hence, the computation of the next time-step can begin with §3.2.1.

### 3.3. Spatial discretisation and filtering

The domain of length  $l_x$  is discretised along the  $x$  axis by a uniform distribution of  $N_x$  nodes  $x_i$ , with  $x_i = (i-1)\Delta x$  for  $1 \leq i \leq N_x$ . In the present algorithm, the spatial derivatives are computed implicitly with high-order finite difference compact schemes. Given a generic function  $f(x)$ , the first derivative  $f'(x)$  is computed with the following expression:

$$\alpha f'_{i-1} + f'_i + \alpha f'_{i+1} = a \frac{f_{i+1} - f_{i-1}}{2\Delta x} + b \frac{f_{i+2} - f_{i-2}}{4\Delta x}. \quad (72)$$

As shown by Lele [50], the coefficients  $\alpha = 1/3$ ,  $a = 14/9$  and  $b = 1/9$  give a quasi-spectral sixth-order accurate approximation of the derivatives. Of course this procedure is the same along the  $y$  and  $z$  directions if the domain is 2D or 3D.

A particular issue concerns the computation with compact schemes of the diffusion operators, which can be defined in a generic way with the following expression:

$$\frac{\partial}{\partial x_i} \left( \mathcal{D} \frac{\partial \vartheta}{\partial x_i} \right), \quad (73)$$

where  $\vartheta$  is the state variable and  $\mathcal{D}$  the associated diffusion coefficient. Two discretisation techniques can be employed to evaluate Eq. (73):

1. by taking the first derivative of  $\vartheta$ , multiplying by  $\mathcal{D}$  and then taking again the first derivative of the whole product;
2. by expanding Eq. (73) with the chain rule, leading to:

$$\frac{\partial \mathcal{D}}{\partial x_i} \frac{\partial \vartheta}{\partial x_i} + \mathcal{D} \frac{\partial^2 \vartheta}{\partial x_i^2}. \quad (74)$$

Under the compact scheme formulation, the second derivative that appears in Eq. (74) is approximated by the following relation:

$$\alpha f''_{i-1} + f''_i + \alpha f''_{i+1} = a \frac{f_{i+1} - 2f_i + f_{i-1}}{2\Delta x^2} + b \frac{f_{i+2} - 2f_i + f_{i-2}}{4\Delta x^2}, \quad (75)$$

where  $f''_i$  represents the second derivative of a function  $f(x)$  at a point  $x_i$ , and where the coefficients  $\alpha = 2/11$ ,  $a = 12/11$  and  $b = 3/11$  have been chosen to ensure sixth-order accuracy with the similar spectral resolution as for the first derivative.

As pointed out by Cook and Riley [23], the second discretisation technique is free of grid-to-grid oscillations but the conservative form of the diffusion operator is not maintained. On the other hand, the first discretisation technique is conservative but develops grid-to-grid oscillations, which can however be removed by spatial filtering.

In the present algorithm, care must be taken in the choice of a discretisation technique. It has been found that the second discretisation technique leads to significant errors in mass conservation when applied to the computation of the diffusion operators  $D_T$  and  $D_{Y_s}$  in the energy equation (6) and species mass fraction transport equation (7). However the first discretisation technique is problematic when applied to the diffusion operator  $D_{u_i}$  in the momentum equation (5), because it introduces spatial oscillations in the velocity field prior to the pressure-projection step. Even with a spatial filtering at this step, numerical inconsistencies have been observed between  $\partial u_i^*/\partial x_i$  and  $\partial u_i^{n+1}/\partial x_i$  during the evaluation of the RHS of Eq. (63), leading to errors in the final velocity field. In summary,  $D_T$  and  $D_{Y_s}$  are evaluated with the discretisation technique 1, while  $D_{u_i}$  is evaluated with the discretisation technique 2.

Periodically at the end of a time-step, a spatial filtering is applied along all directions to the state variables  $T$ ,  $Y_s$  and  $u_i$ . According to Lele [50], the spatial filtering operator can be done with the help of compact filters by applying the following relation:

$$\alpha \bar{f}_{i-1} + \bar{f}_i + \alpha \bar{f}_{i+1} = a f_i + \frac{b}{2} (f_{i+1} + f_{i-1}) + \frac{c}{2} (f_{i+2} + f_{i-2}) + \frac{d}{2} (f_{i+3} + f_{i-3}), \quad (76)$$

where  $\bar{f}$  is the filtered variable while the following coefficients represent a sixth-order low-pass filter:

$$a = \frac{1}{16} (11 + 10\alpha), \quad (77)$$

$$b = \frac{1}{32} (15 + 34\alpha), \quad (78)$$

$$c = \frac{1}{16} (-3 + 6\alpha), \quad (79)$$

$$d = \frac{1}{32} (1 - 2\alpha). \quad (80)$$

Note that  $-0.5 < \alpha < 0.5$  is a parameter freely set by the user to control the spectrum of the filter. A value of  $\alpha = 0.5$  implies no filtering, while reducing  $\alpha$  moves the cut-off frequency to low-band values. Note that a value of  $\alpha = 0.48$  has been chosen for all the simulations conducted in the present paper.

When non-periodic boundary conditions are imposed, single sided formulations are used for the approximation of the first and second derivatives. For the boundary nodes, the derivatives have the form

$$f'_1 + 2f'_2 = \frac{1}{2\Delta x} (-5f_1 + 4f_2 + f_3), \quad (81)$$

$$f''_1 + 11f''_2 = \frac{1}{\Delta x^2} (13f_1 - 27f_2 + 15f_3 - f_4), \quad (82)$$

that are third-order accurate, while at adjacent nodes the following fourth-order accurate formulations are used:

$$\frac{1}{4}f'_1 + f'_2 + \frac{1}{4}f'_3 = \frac{3}{2} \frac{f_3 - f_1}{2\Delta x}, \quad (83)$$

$$\frac{1}{10}f''_1 + f''_2 + \frac{1}{10}f''_3 = \frac{6}{5} \frac{f_3 - 2f_2 + f_1}{\Delta x^2}. \quad (84)$$

For spatial filtering with compact schemes, several formulations for points near boundaries can be found in the review paper of Gaitonde and Visbal [51].

Note that the pressure is discretised on a staggered grid and interpolations and derivatives are performed at mid-points via sixth-order compact schemes. Concerning the FFT solver, the transposition between the physical space and the spectral space is performed under the so-called *spectral equivalence principle*. Basically, this means that derivations/interpolations in the physical space lead strictly to the same results in the spectral space. Recall that the present **HOLOMAC** code is built upon the **Incompact3D** framework and all of these procedures are discussed in detail in the reference paper [41].

### 3.4. Extrapolation/interpolation of thermodynamic properties

The temporal integration of diffusion operators  $D_{u_i}$ ,  $D_T$ ,  $D_{Y_s}$  through the RKC stages requires the knowledge of thermodynamic properties between two computational time-steps, say  $t^n$  and  $t^{n+1}$ . Computing these properties with the **CHEMKIN** libraries is computationally costly. Repeated evaluation within the RKC stages would drastically increase the computational burden. Najm and Knio [37] proposed to estimate the intermediate values via extrapolation and interpolation procedures. It was also demonstrated that such approaches recover an overall second-order accuracy in time. Thus, the direct evaluation of the properties with **CHEMKIN** libraries is only performed once for each time-step at the end of step §3.2.4.

Denoting  $t^{RKC}$  as the intermediate time, i.e.  $t^n \leq t^{RKC} < t^{n+1}$ , the generic thermodynamic parameter  $\psi$  is evaluated as follows:

- In steps §3.2.2 and §3.2.4, values at time  $t^{RKC}$  are extrapolated with known values at  $t^n$  and  $t^{n-1}$ , i.e.

$$\psi^{RKC} = \psi^n + \frac{t^{RKC} - t^n}{\Delta t^n} (\psi^n - \psi^{n-1}), \quad (85)$$

where  $\psi$  refers to  $\mathcal{D}_s$ ,  $C_p$  and  $\lambda$ .

- In step §3.2.6, values at time  $t^{RKC}$  are interpolated from known values at  $t^n$  and  $t^{n+1}$ , i.e.

$$\psi^{RKC} = \psi^n + \frac{t^{RKC} - t^n}{\Delta t^n} (\psi^{n+1} - \psi^n), \quad (86)$$

where  $\psi$  refers to  $\mu$  and  $\rho$ .

## 4. Results

In this section the performance of the numerical methods developed in Sec. 3 is assessed. First, temporal and spatial convergence tests are performed for a freely propagating 1D methane/air flame. Two configurations are investigated:

- an open domain, where unburnt fuel/air mixture enters the domain at the intake boundary with the flame speed;
- a periodic closed domain where two flames propagate in opposite directions into the unburnt mixture from a hot spot.

Both cases exhibit an overall second-order accuracy in time, validating the implementation of a Strang operator-split strategy and the pressure-projection procedure into a fractional-step algorithm. As expected, the algorithm in a periodic closed domain shows sixth-order accuracy in space, while the use of inflow/outflow boundary conditions in a open domain gives an overall spatial accuracy of order 4.5, due to the degradation of the precision of compact schemes near the boundaries. Following on, a second test problem concerning a 2D vortex interacting with a flame is reported in §4.2.

Another test problems are employed in §4.3 to evaluate the three methods presented in §3.2.7 to solve the variable-coefficient Poisson equation. In a first test problem, a heptane/air mixing layer is established and the autoignition of the mixtures is investigated in a 1D closed domain. In this problem, time-evolving strong gradients of the physical variables are generated, leading to difficulties in the convergence of the pressure equation. It is shown that method **III** efficiently achieves machine precision accuracy while maintaining a low number of iterations when solving the variable-coefficient Poisson equation for the pressure. Following on, the same study is extended to the 2D vortex/flame interaction test case.

In §4.4, numerical experiments are performed with the RKC method to study the influence of the number of integration stages on the stability behaviour. Finally, in §4.5, 3D simulations of a premixed turbulent flame in a lean methane/air mixture are reported, demonstrating the ability of the numerical algorithm to handle complex turbulent reacting flows.

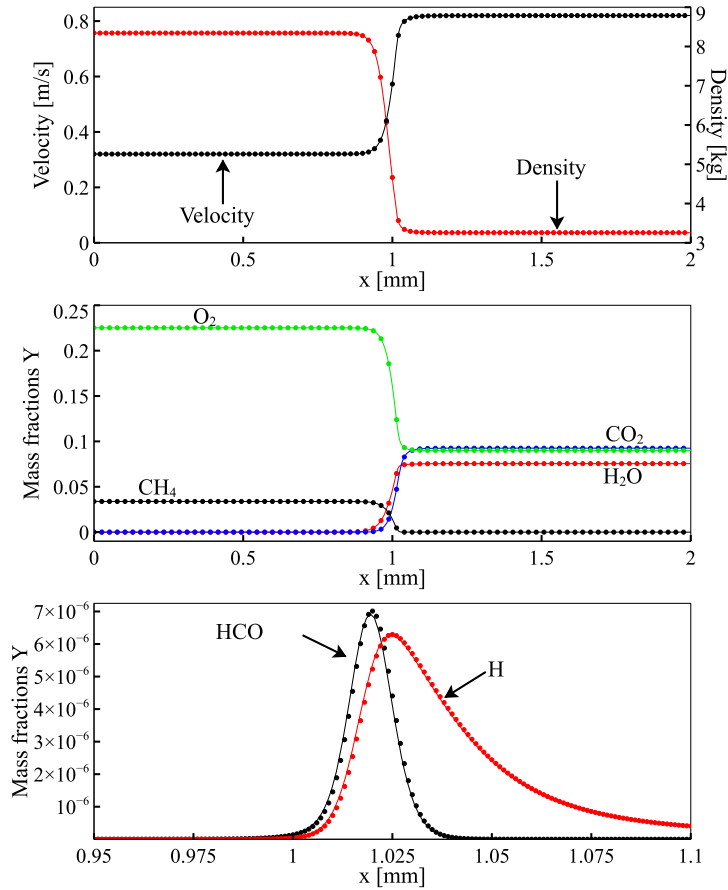
In the remainder of the paper, results computed with the numerical methods detailed in Sec. 3 are referred by **HOLOMAC**, which is the name of the software developed during the present study to perform the numerical simulations.

### 4.1. Temporal and spatial convergence tests

#### 4.1.1. One-dimensional freely propagating premixed methane flame in an open domain

The physical problem consists of a freely propagating premixed methane ( $\text{CH}_4$ )/air flame in a one-dimensional domain. All flames are computed with the skeletal mechanism for lean methane–air developed by Sankaran et al. [52], composed of 17 species and 73 reactions. Moreover, the fresh gas is a premixed mixture of methane and air, with a temperature set to 810 K and an equivalence ratio of  $\Phi = 0.6$ . Hence,  $Y_{\text{CH}_4}^0 = 0.0338$ ,  $Y_{\text{O}_2}^0 = 0.2252$  and  $Y_{\text{N}_2}^0 = 0.741$ . In the present simulation, the profile of a 1D methane/air premixed flame is computed with the reference code **CHEMKIN** [39], and imposed as an initial condition for the simulation with **HOLOMAC**. The relative and absolute tolerances for the **CHEMKIN** computation are set to  $10^{-12}$  and  $10^{-8}$ , respectively. The total length of the computational domain is 0.04 m, and the mesh grid is progressively refined with an adaptative technique. At convergence, the solution is discretised on approximately 9000 points and the computed flame speed is  $u_f = 0.3202 \text{ ms}^{-1}$ . For this configuration, the thermodynamic pressure  $p_0$  is set to  $2 \times 10^6 \text{ Pa}$ . The flame is kept stationary in the domain by setting the inflow velocity  $u_f$  to be the flame speed.

The premixed flame profile is interpolated from **CHEMKIN** into **HOLOMAC** on a domain of 2 mm, the flame front being located at 1 mm. In order to ensure a consistent initial condition to perform convergence test, a preliminary computation is performed to let the flame profile adapt to the mesh grid of the new solver. As the influence of numerical errors must be as small as possible, the domain is discretised with 2001 points, which represents a size between each grid points of  $\Delta x \approx 1 \mu\text{m}$ . Moreover, a very small time-step of  $\Delta t = 1 \times 10^{-8} \text{ s}$ , which corresponds to a convective CFL number of 0.008, is set to minimise splitting errors between convection–diffusion–reaction operators. The absolute and relative tolerances for the DVODE solver are both set to  $10^{-14}$ , while method **III** is used to solve the variable-coefficient Poisson equation with a tolerance parameter  $\xi$  set to  $10^{-10}$ . The number of stages for the RKC method is set to  $K = L = 10$  for the integration of diffusion terms. Note that for an explicit integration of the diffusion terms without the RKC method, the von Neumann stability criterion for the diffusion would impose a theoretical maximum time-step of approximately  $3.5 \times 10^{-12} \text{ s}$ , which is four orders below the selected time-step. At 1 ms of the physical simulation time, a solution  $S_{\text{init}}$  is stored.



**Fig. 1.** Structure of a steady one-dimensional premixed methane/air flame at  $2 \times 10^6$  Pa and equivalence ratio of  $\Phi = 0.6$ . Solid line (—): initial solution interpolated from a **CHEMKIN** simulation. Symbols (•): solution  $S_{init}$  computed at 1 ms with **HOLOMAC** on a fine mesh ( $\Delta x = 1 \mu\text{m}$ ) and a very small time-step ( $\Delta t = 1 \times 10^{-8}$  s).

The typical structure of the computed premixed methane flame is depicted in Fig. 1. The solid line represents the initial solution interpolated from the **CHEMKIN** simulation, while the dot symbols represent the solution  $S_{init}$  computed with **HOLOMAC**. Differences are very small. Note that the temperature, while not depicted, follows a trend similar to the velocity profile. The temperature of unburnt and burnt mixtures are 810 K and 2073 K, respectively. As shown in the bottom of Fig. 1, the influence of numerical errors is visible in the computation of radical species, for example typically  $\mathcal{O}(10^{-6})$  for  $Y_{\text{HCO}}$  and  $Y_{\text{H}}$ . These results validate the ability of the algorithm and the numerical methods developed in the present paper to accurately simulate a reacting flow with detailed chemistry.

Convergence tests are evaluated with the  $\mathcal{L}^2$ -norm of the difference between the computed and the reference solutions, which is expressed as follows:

$$\mathcal{L}^2(S_{sol} - S_{ref}) = \sqrt{\frac{(\phi_{sol} - \phi_{ref})^2}{N_x}}, \quad (87)$$

where subscripts *sol* and *ref* identify the computed and reference solutions,  $\phi$  is the variable investigated, and  $N_x$  is the number of points of the mesh grid. The convergence rate is then computed by best-fitting the curve formed by successive  $\mathcal{L}^2$ -norms.

Recall that  $S_{init}$  is the solution computed on the 2001 points grids ( $\Delta x \approx 1 \mu\text{m}$ ) with a time-step of  $1 \times 10^{-8}$  s at 1 ms after the **CHEMKIN** profile is fitted on the grid. The procedures to perform convergence tests are as follows:

- for the spatial accuracy, the solution  $S_{init}$  is taken as the reference  $S_{ref}$ . The time-step is kept at  $1 \times 10^{-8}$  s and simulations are performed over a physical time of 0.5 ms on successive mesh grids of 1001, 501, 251 and 151 points, which represents  $\Delta x \approx 2, 4, 8, 13.25 \mu\text{m}$ , respectively;
- for the temporal accuracy, the reference solution  $S_{ref}$  is formed by interpolating  $S_{init}$  on a mesh grid of 1001 points ( $\Delta x \approx 2 \mu\text{m}$ ). Simulations are performed over a physical time of 0.1 ms with successive time-steps of  $\Delta t = 2, 3.125, 4, 6.25, 8 \times 10^{-8}$  s.

**Table 1**

Spatial and temporal global rates of convergence for the premixed 1D methane/air flame in an open domain.

Variable	Global rates of convergence	
	Spatial	Temporal
$T$	4.6	2.27
$u$	4.57	2.27
$p_1$	4.52	2.18
$\rho$	4.78	2.27
$Y_{\text{CH}_4}$	4.45	2.30
$Y_{\text{O}_2}$	4.54	2.27
$Y_{\text{CO}_2}$	4.63	2.22
$Y_{\text{CO}}$	3.84	2.33
$Y_{\text{H}_2\text{O}}$	4.55	2.29
$Y_{\text{OH}}$	3.63	2.17
$Y_{\text{CH}_3}$	3.25	2.23
$Y_{\text{HCO}}$	2.81	1.60
$Y_{\text{H}}$	3.30	1.99

**Table 2**

Partial temporal convergence rates over a selected range of time-steps for the premixed 1D methane/air flame in an open domain.

$\Delta t$ range (s)	$T$	$u$	$p_1$	$\rho$	$Y_{\text{CH}_4}$	$Y_{\text{HCO}}$
$(4 \rightarrow 6.25 \rightarrow 8) \times 10^{-8}$	1.66	1.66	1.65	1.66	1.66	1.31
$(3.125 \rightarrow 4 \rightarrow 6.25) \times 10^{-8}$	2.06	2.06	2.04	2.06	2.08	1.44
$(2 \rightarrow 3.125 \rightarrow 4) \times 10^{-8}$	2.40	2.40	2.33	2.40	2.43	1.63
$(1 \rightarrow 2 \rightarrow 3.125) \times 10^{-8}$	2.76	2.76	2.52	2.78	2.83	1.97

Results are gathered in Table 1 for a selection of variables. The global spatial rate of convergence is approximately  $\mathcal{O}(4.5)$ . This result was expected, because as described in §3.3 the order of accuracy of the spatial discretisation schemes is degraded to third-order at boundary points and fourth-order accuracy for adjacent points. Due to the implicit nature of compact schemes, despite the sixth-order accuracy for the remaining interior nodes, this degradation at boundaries introduces then an error  $\mathcal{O}(3)$  in the whole domain that impede the global spatial rate of convergence. With a balance between sixth-order and third-order accuracy for the discretisation schemes, an overall accuracy  $\mathcal{O}(4.5)$  can be expected, which is confirmed by results presented in Table 1. As explained above, species with very small mass fractions are more sensitive to numerical errors and it is observed that the global spatial convergence rate for such variables can be affected and reduced to approximately  $\mathcal{O}(3)$ .

Furthermore, results reported in Table 1 show a temporal global rate of convergence of order 2. This result was expected and it validates that an implementation of a Strang operator-split strategy and a pressure-projection procedure into a fractional-step algorithm ensures a second-order accuracy in time. One can notice that the global temporal convergence rate of  $Y_{\text{HCO}}$ , an intermediate species of low concentration, is only 1.60. Table 2 presents the temporal convergence rates for a selected ranges of time-steps. These results reveal that the order of accuracy increases as  $\Delta t$  is reduced. Moreover, the temporal convergence rate of  $Y_{\text{HCO}}$  reaches second-order accuracy slower than other selected variables. This behaviour may be explained by the fact that HCO is a species sensitive to the evaluation of chemical reactions, and is more prone to be affected by errors introduced by the operator-split procedure. Of course such statement depends on the mechanism employed for chemistry and the configuration of the flow. In the context of methane/air flame, Safta et al. [27] also observed, with a similar chemistry mechanism (16 species and 46 reactions), the same behaviour for the temporal convergence of  $Y_{\text{HCO}}$ .

#### 4.1.2. One-dimensional freely propagating premixed methane flame in a periodic closed domain

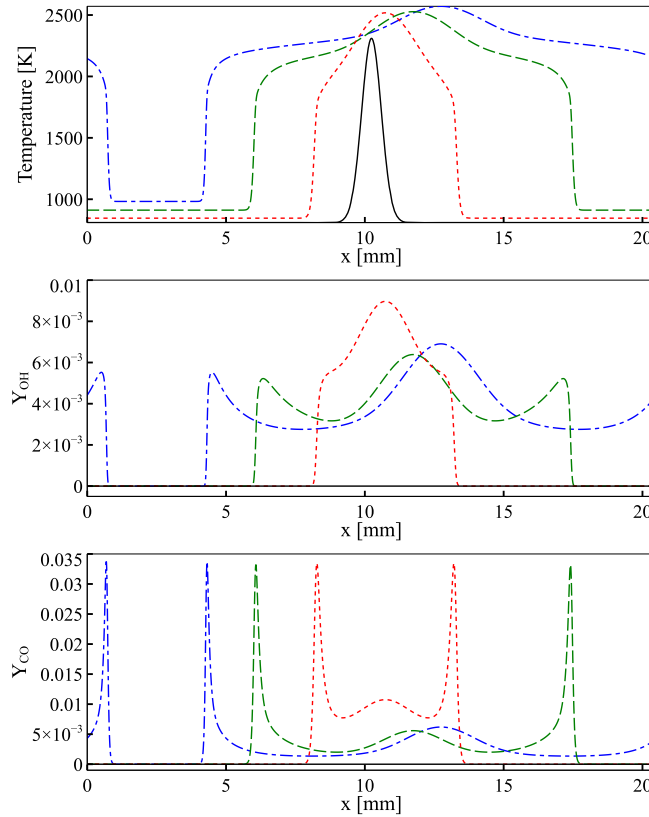
This test case is devoted to assess the spatial and temporal rate of convergence of the numerical methods when the domain is fully closed by periodic boundary conditions. Moreover, unlike the test case investigated previously in §4.1.1, the time derivative of the thermodynamic pressure (see Eqs. (27) and (55)) is now involved in the energy equation (6) and the divergence constraint expressed by Eq. (54). Thus,  $p_0$  is expected to evolve. Although not detailed in the present paper, the algorithm with time evolving  $p_0$  has been validated against CHEMKIN with a test case of autoignition of a homogeneous heptane/air mixture in a periodic, closed domain.

The initial solution  $S_{\text{init}}$  is obtained for the fresh premixed mixture of methane and air of equivalence ratio of  $\Phi = 0.6$  as described at the beginning of Sec. 4.1. The imposed temperature follows a Gaussian profile:

$$T = T_{\text{fresh}} + T_{\text{amp}} \exp\left(-\left(\frac{1 - l_x/2}{\delta}\right)^2\right) \quad (88)$$

where  $T_{\text{fresh}} = 810$  K is the temperature of the unburnt mixture and  $T_{\text{amp}} = 1500$  K is the maximum amplitude of the Gaussian,  $l_x = 20.48$  mm being the total length of the computational domain and  $\delta = 0.5$  mm the width of the Gaussian.





**Fig. 2.** Structure of a freely propagating one-dimensional premixed methane/air flame in a closed domain at 101325 Pa and equivalence ratio of  $\Phi = 0.6$ . Top: temperature. Middle:  $Y_{OH}$ . Bottom:  $Y_{CO}$ . Solid line (—) initial solution at  $t = 0$  ms. Dotted line (---):  $t = 0.50$  ms. Dashed line (---):  $t = 1.50$  ms. Dashed-dotted line (---):  $t = 2.50$  ms.

The initial temperature profile is depicted at the top of Fig. 2. Moreover, a velocity of  $1 \text{ ms}^{-1}$  is uniformly imposed on the flow.

Fig. 2 presents the flame structure for three physical times:  $t = 0.50$  ms,  $t = 1.50$  ms and  $t = 2.50$  ms. Very shortly after the beginning of the simulation, the deposition of a hot temperature spot ignites the mixture, creating then two flame fronts propagating in opposite directions. Although not shown, the two flames merge at approximately  $t = 3$  ms: the fresh premixed mixture is then totally consumed and only burnt gas remains in the domain.

The procedures to perform convergence tests are described as follows:

- for the spatial accuracy, the time-step is set to  $1 \times 10^{-7}$  s and simulations are performed over a physical time of 2.50 ms on successive mesh grids of 2048, 1536, 1024, 768, 512 and 384 points, which represent  $\Delta x \approx 10, 13.3, 20, 26.6, 40, 53.3 \mu\text{m}$ , respectively;
- for the temporal accuracy, a mesh grid of 1024 points ( $\Delta x = 20 \mu\text{m}$ ) is used. Simulations are performed over a physical time of 2.50 ms with successive time-steps of  $\Delta t = 1, 2, 4, 8, 12.5, 20 \times 10^{-7}$  s.

Note that for all computations performed to assess the convergence rates, the absolute and relative tolerances for the DVODE solver are both set to  $10^{-14}$ , while method III is used to solve the variable-coefficient Poisson equation with a tolerance parameter  $\xi$  set to  $10^{-10}$ . The number of stages for the RKC method is set to  $K = L = 12$  for the integration of diffusion terms. This value ensures stability for the computation, which otherwise would become unstable for the largest time-steps (with  $\Delta t = 20 \times 10^{-7}$  s and  $\Delta x = 20 \mu\text{m}$ ). Indeed, for this configuration the von Neumann stability criterion for the diffusion would impose a theoretical maximum time-step of approximately  $1.2 \times 10^{-9}$  s. Moreover the CFL number reaches 0.7 during the autoignition of the flame when the gases expand rapidly in either direction from the reacting zone. The CFL number of 0.7 is near the theoretical stability limit for explicit time integration.

Results are gathered in Table 3 for a selection of variables. The global spatial and temporal rates of convergence are approximately sixth-order and second-order, respectively. As the spatial discretisation is performed with sixth-order compact schemes everywhere in the computational domain, this result was expected. Moreover the addition of the time derivative of thermodynamic pressure into the energy equation and the divergence constraint does not affect the temporal accuracy.

**Table 3**  
Spatial and temporal global rates of convergence for the premixed 1D methane/air flame in a periodic, closed domain.

Variable	Global rates of convergence	
	Spatial	Temporal
$T$	5.91	2.11
$u$	5.97	1.86
$p_1$	7.48	1.93
$\rho$	6.24	2.10
$Y_{\text{CH}_4}$	5.98	2.12
$Y_{\text{O}_2}$	5.89	2.11
$Y_{\text{CO}_2}$	5.99	1.98
$Y_{\text{CO}}$	5.81	2.07
$Y_{\text{H}_2\text{O}}$	5.98	2.11
$Y_{\text{OH}}$	5.73	1.97
$Y_{\text{CH}_3}$	5.34	1.99
$Y_{\text{HCO}}$	5.23	1.70
$Y_{\text{H}}$	5.61	1.95

Note that like the test case investigated in §4.1.1, the same issues for mass fractions of small amplitudes such as  $Y_{\text{HCO}}$  are observed in terms of convergence rates.

All convergence tests performed within this section used method III with a tolerance parameter  $\xi$  set to  $10^{-10}$  to solve the variable-coefficient Poisson Eq. (63) up to machine precision accuracy. The properties of the three methods proposed in §3.2.7 as well as the influence of the tolerance parameter  $\xi$  are now investigated in the following section.

#### 4.2. 2D vortex-flame interaction

This test case aims to simulate the interaction between a pair of vortices and a flame front. Similar cases have been studied in the literature but with different fuel, flow configuration and numerical methods [53,26,37,54,28]. As recalled by Lessani and Papalexandris [54], quantitative comparisons are difficult to establish because the aforementioned works do not provide sufficient information to exactly reproduce the same flame. However, this case represents a good numerical test to assess the implementation in 2D, as well as the stability behaviour of the code when vortices are leaving the domain through the outflow boundary.

The simulations are carried out in a rectangular domain of length  $l_x = 4$  mm and height  $l_y = 2$  mm. The freely propagating 1D methane/air flame, which has been previously computed in §4.1.1, is imported as an initial condition, and flow parameters as well as boundary conditions are unchanged. The position of the flame front is located at  $x = 1$  mm, and the flame profile is homogeneously recopied along the  $y$  axis.

Two vortices are superimposed on the flow. One vortex can be described by the following expressions:

$$u_x = u_f + C \frac{y - y_0}{R^2} \exp\left(-r^2/2\right), \quad u_y = -C \frac{x - x_0}{R^2} \exp\left(-r^2/2\right), \quad (89)$$

where  $u_f$  is the initial velocity profile of the flame along the  $x$  axis, while  $x_0$  and  $y_0$  are the coordinates of the center of a vortex and  $C$  and  $R$  denote the strength and radius, respectively. Finally,  $r = \sqrt{((x - x_0)^2 + (y - y_0)^2)}/R$ , and in the present simulation  $R = l_y/48$ . If the subscripts 1 and 2 refer to the clockwise and counterclockwise rotating vortices, respectively, their initial locations and parameter  $C$  are defined by

$$x_{01} = l_x/4, \quad y_{01} = l_y/2 - l_y/6, \quad C_1 = +0.3 \times 10^{-3}, \quad (90)$$

$$x_{02} = l_x/4, \quad y_{02} = l_y/2 + l_y/6, \quad C_2 = -0.3 \times 10^{-3}. \quad (91)$$

The inflow/outflow configuration is problematic, especially for the treatment of the hydrodynamic pressure. Indeed, as a zero pressure gradient is imposed at inflow and outflow boundaries, fluctuations leaving the domain through the outflow boundary may trigger non-physical oscillations of the hydrodynamic pressure in the whole domain, leading to a destabilisation of the simulation. It is well known that the design of a non-reflecting radiation condition at the outlet of a computational domain is difficult, especially in 2D/3D. There is an extensive literature on this subject but there is no consensus on resolving the challenges, and the proposition of a general form for open conditions is still under investigation [55]. In the present code, the 2D extension of the Orlanski outflow boundary condition proposed by Raymond and Kuo [56] was tested, but no real improvement in the stability was observed.

Furthermore, this stability problem can be efficiently overcome by imposing a sponge layer zone near the outlet, so as to damp fluctuations and virtually keep the flow variables constant at the boundary. The method proposed by Billson et al. [57] was implemented and successfully tested. Basically a source term is added to the governing equations:

$$\frac{\partial \mathbf{Q}}{\partial t} = \dots - \frac{\sigma(x)}{\Delta x} (\mathbf{Q}^n - \bar{\mathbf{Q}}^n), \quad (92)$$

where

$$\sigma(x) = \sigma_{\max} \left( \frac{x - x_0}{x_{\max} - x_0} \right)^2, \quad (93)$$

and

$$\bar{\mathbf{Q}}^n = \varpi \bar{\mathbf{Q}}^{n-1} + (1 - \varpi) \mathbf{Q}^n. \quad (94)$$

In the above equations,  $\mathbf{Q}$  represents the solution vector and  $\sigma_{\max} = 1$  in the present simulation, while the beginning and the end of the sponge region is bounded by  $x_0 = 3$  mm and  $x_{\max} = 4$  mm, respectively. The term  $\bar{\mathbf{Q}}^n$  represents a target field where the computed solution should approach. A value can be provided, but it is however difficult to guess. In the present simulation, this term is constructed by time-averaging of the solutions over the first 50 time-steps. During this process,  $\varpi = 0.99$ . For the remaining of the simulation,  $\varpi = 1$  so as to deactivate the averaging process. Note that in the present simulation, the damping term described at Eq. (92) is only applied to the momentum and temperature equations (5)–(6), the species mass fractions being not directly constrained but following the density and temperature evolution.

The present simulation was performed on a regular, cartesian mesh grid of  $1024 \times 512$  points in the  $x$  and  $y$  directions, respectively. A constant  $\Delta t = 20$   $\mu$ s was selected, whereas the number of RKC stages were set to  $K = L = 5$ . Both absolute and relative tolerances were set to  $1 \times 10^{-8}$  for the implicit integration of the chemistry with DVODE, while  $\xi = 1 \times 10^{-6}$  was set for the resolution of the Poisson equation. The simulation was run over a physical time of 3 ms on 192 CPUs, which represent a wall-clock time of 120 hours.

Results are shown in Fig. 3 at 4 instants after start of the computation. The mass fraction of the species OH is employed to distinguish the flame front. The vorticity is depicted by the contour lines, the white ones and black ones representing the clockwise and counterclockwise rotations, respectively. At the beginning of the simulation, the vortices are naturally convected by the flow inside the fresh mixture. As they come closer to the reaction zone, the reaction front is stretched and flame-generated vorticity appears along the flame front. When crossing the flame, the initial vortices are strongly dissipated, and then convected away from the flame by the burnt mixture. This physical behaviour is qualitatively similar to other simulations reported in the aforementioned references. Moreover, as evident in the shapes of the flame front, the symmetric features of the flow with respect to the central axis at  $l_y/2$  are perfectly reproduced, verifying then the 2D implementation.

Note that the present case was also run with a coarser mesh of  $512 \times 256$  points in the  $x$  and  $y$  directions, respectively. The simulation was advanced in time over 7 ms so as to assess the stability of the simulation when the vortices leave the computational domain. Although not shown in the present paper, results show that computations with the sponge layer region is efficient and that no instabilities arise either from the outflow boundary, or from the interface between the beginning of the damping zone and the physical computational domain.

#### 4.3. Variable-coefficient Poisson equation: assessment of resolution strategies

The present section is devoted to the analysis of the three methods proposed in §3.2.7 to solve the variable-coefficient Poisson equation (63), as well as the influence of the tolerance parameter  $\xi$ . This study relies first on a test case of autoignition of mixing layers of heptane ( $C_7H_{16}$ ) and air in a domain closed by solid walls. Then the study is extended to the 2D Vortex-Flame configuration previously reported in §4.2.

##### 4.3.1. One-dimensional autoignition of a heptane–air mixture in a closed domain

The configuration consists of a one-dimensional domain of length  $l = 5$  mm. Walls are imposed at both ends, and the temperature and species mass fractions evolve freely, i.e.  $u = 0$ ,  $\partial T / \partial x = 0$  and  $\partial Y_s / \partial x = 0$ . The initial thermodynamic pressure  $p_0^0$  is set to  $4 \times 10^6$  Pa, while the mixing layers have an initial uniform temperature of  $T^0 = 1000$  K and follow a species profile given by

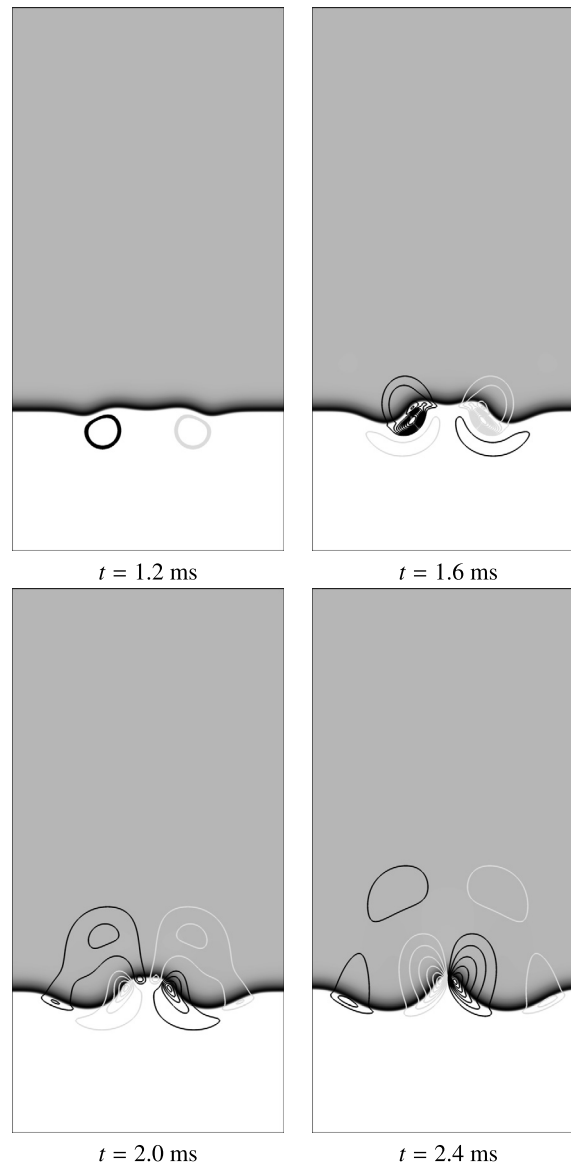
$$Y_{C_7H_{16}}^0 = \frac{1}{2} (1 - \tanh((x - x_f) / \delta_f)), \quad (95)$$

$$Y_{O_2}^0 = 0.233 \left( 1 - \frac{1}{2} (1 - \tanh((x - x_f) / \delta_f)) \right), \quad (96)$$

$$Y_{N_2}^0 = 1 - Y_{C_7H_{16}}^0 - Y_{O_2}^0, \quad (97)$$

where  $x$  is the coordinate of a mesh point, while  $x_f = l/2$  and  $\delta_f = 120$   $\mu$ m are the location and the thickness of the interface between heptane and air, respectively. Moreover, no initial velocity is imposed on the flow, hence  $u^0 = 0$  everywhere.

The mesh grid is composed of 1001 points, which represents a space grid of  $\Delta x \approx 5$   $\mu$ m. The time-step is set to  $\Delta t = 1 \times 10^{-7}$  s and the simulations are performed over a physical time of 0.3 ms. The number of stages for the RKC integration of diffusions terms are set to  $K = L = 5$ . Note that the von Neumann stability criterion for the diffusion would impose a theoretical maximum time-step of approximately  $7 \times 10^{-11}$  s. The absolute and relative tolerance parameters for DVODE are both set to  $10^{-14}$ . Moreover in the present numerical set-up, the convective CFL reaches a maximum value of 0.02 at 0.2 ms

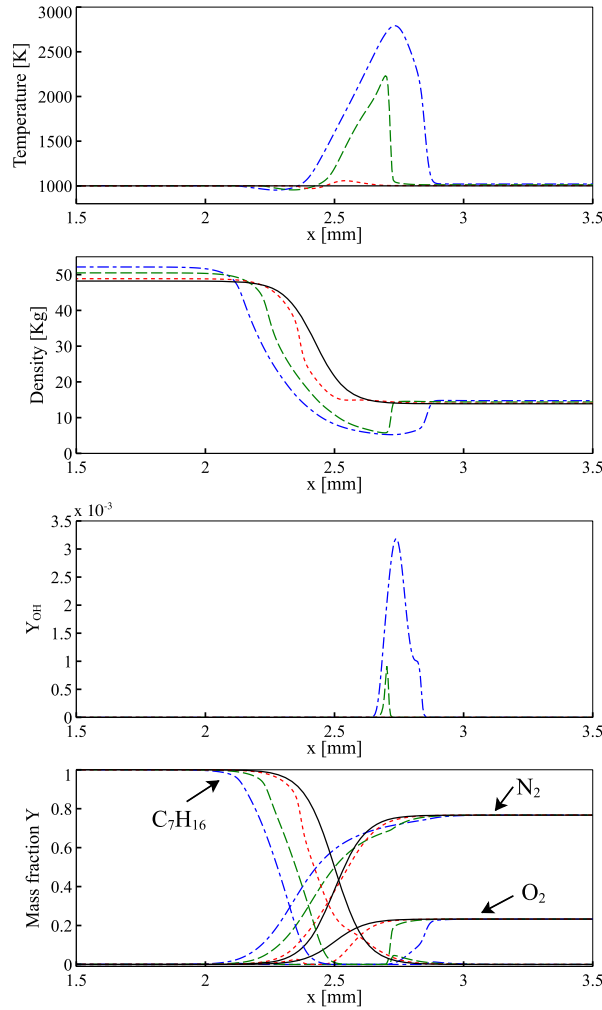


**Fig. 3.** Simulation of the interaction of a vortices pair with a methane/air flame at  $t = 1.2$  ms,  $t = 1.6$  ms,  $t = 2.0$  ms and  $t = 2.4$  ms. The mass fraction of the species OH is represented to distinguish the flame front. The vorticity is depicted by the contour lines, the white ones and black ones representing the clockwise and counterclockwise rotations, respectively.

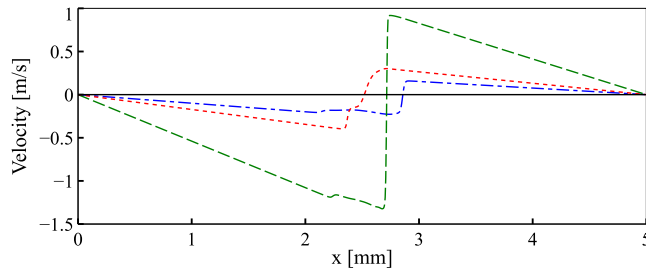
when the velocity is maximum (see below). Such a low value is selected to ensure that no stability issues will arise during the study of the methods proposed to solve the variable-coefficient Poisson equation. Finally, the chemistry is described by a 37 species, 56 reactions mechanism proposed by Peters et al. [58].

The temporal evolution of the flow is depicted in Fig. 4 for a selection of variables and for three times during the autoignition process. From the beginning of the simulation until  $t = 0.1$  ms, the fuel (heptane) and the air are mixing through laminar diffusion. As the initial pressure and temperature are high, typical of diesel combustion engines, the mixture then auto-ignites and the temperature rises rapidly to values higher than 2000 K. Hence, as shown in Fig. 5 for  $t = 0.2$  ms, a strong gradient of velocity is generated due to expansion. By about 0.3 ms the temperature has stabilised at approximately 2800 K, and the peak velocity decreases to approximately  $0.15 \text{ m s}^{-1}$ .

The methods proposed in §3.2.7 are now investigated. The simulation of autoignition of heptane is repeatedly performed with methods I, II and III. For methods I and III, the tolerance parameter is varied with values  $\xi = 1, 2, 4, 6, 8, 10, 12 \times 10^{-12}$ , method II being without iterations. The solutions are the velocity fields taken at  $t = 3$  ms and Eq. (87) is used to compute the  $\mathcal{L}^2$ -norm of the errors formed with the reference solution taken to be the solution obtained with method I and with  $\xi = 10^{-12}$ . This choice is justified by the fact that for  $\xi = 10^{-12}$ , both methods I and III give virtually the same solution.



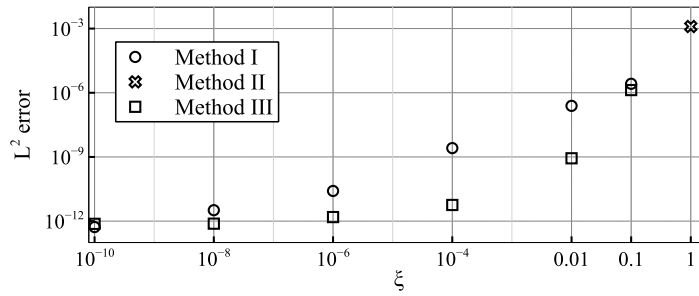
**Fig. 4.** Structure of an autoignition sequence of a one-dimensional flame into mixing layers of heptane and air, at  $4 \times 10^6$  Pa. Solid line (—) initial solution at  $t = 0$  ms. Dotted line (· · · · ·):  $t = 0.1$  ms. Dashed line (— — —):  $t = 0.2$  ms. Dashed-dotted line (— · — ·):  $t = 0.3$  ms.



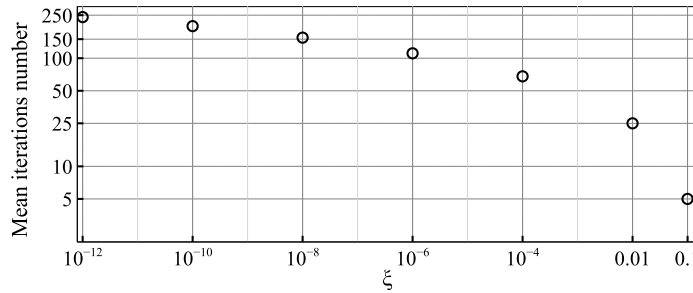
**Fig. 5.** Velocity of the flow during an autoignition sequence of a one-dimensional flame into mixing layers of heptane and air, at  $4 \times 10^6$  Pa. Solid line (—) initial solution at  $t = 0$  ms. Dotted line (· · · · ·):  $t = 0.1$  ms. Dashed line (— — —):  $t = 0.2$  ms. Dashed-dotted line (— · — ·):  $t = 0.3$  ms.

Results are shown in Fig. 6. The methods I, II and III are represented by circle, cross and square symbols, respectively. Obviously method II is the least accurate as the error is  $\mathcal{O}(10^{-3})$ , which is far from the machine precision limit. Both methods I and III converge to the machine precision accuracy for very low  $\xi$  values. However an interesting feature of method III is that the accuracy of  $\mathcal{O}(10^{-12})$  is reached for values of  $\xi$  of  $\mathcal{O}(10^{-6})$ , while with method I such order of accuracy is only reached for  $\xi$  of  $\mathcal{O}(10^{-12})$ . This suggests that method III significantly accelerates the convergence when the pressure equation is solved iteratively.

As visible in Fig. 4 for the density and Fig. 5 for the velocity, there are strong gradients in the flow that evolve quickly in time, and with a density ratio of about 11. Hence, due to the dependence of the non-linear Poisson equation on the density,



**Fig. 6.**  $\mathcal{L}^2$  norm errors of the velocity field at  $t = 3$  ms formed by methods **I**, **II** and **III** respect to the tolerance parameter  $\xi$  for the heptane/air autoignition test case.



**Fig. 7.** Mean number of iterations performed to reach the specified tolerance  $\xi$  for the heptane/air autoignition test case when methods **I** and **III** are chosen to solve the variable-coefficient Poisson equation (63).

its resolution up to machine accuracy is difficult and require a significant computational effort. Indeed, as shown in Fig. 7, both methods **I** and **III** require the same number of approximately 250 iterations to solve the pressure Poisson equation up to machine accuracy. However choosing  $\xi = 10^{-6}$  with method **III** requires about 100 iterations to reach the accuracy of  $\mathcal{O}(10^{-12})$  during the projection-correction step. Note that the numbers of iterations presented in Fig. 7 are mean values computed over a whole simulation. It is also worth noting that this test case has been chosen to represents a flow under severe conditions, i.e. with transient ignition. In the context of the freely propagating methane/air flames presented in §4.1, solving the pressure Poisson equation with method **III** and  $\xi = 10^{-10}$  requires only approximately 30 iterations.

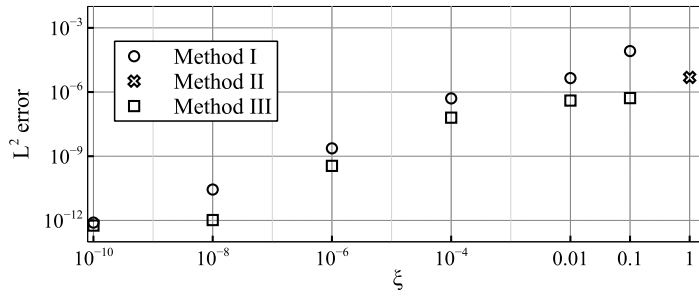
#### 4.3.2. Extension to the 2D vortex-flame test case

The simulation of the interaction between a pair of vortices and a premixed methane flame, previously reported in §4.2, is repeatedly performed with methods **I**, **II** and **III**. Similarly to the 1D study presented in §4.3.1, for methods **I** and **III**, the tolerance parameter is varied with values  $\xi = 1, 2, 4, 6, 8, 10, 12 \times 10^{-12}$  and there are no iterations for method **II**. The solutions are the velocity fields taken at  $t = 1$  ms when the pair of vortices interacts with the flame front. Equation (87) is used to compute the  $\mathcal{L}^2$ -norm of the errors formed with the reference solution taken to be the solution obtained with method **I** and with  $\xi = 10^{-12}$ .

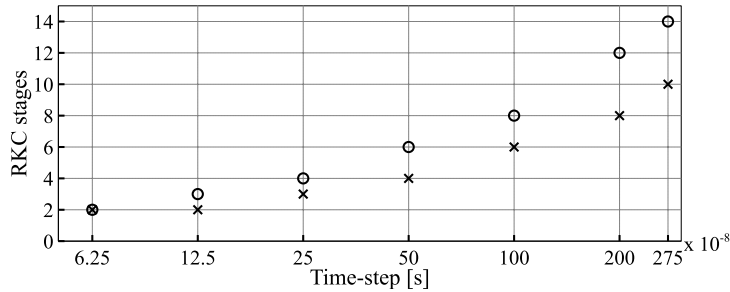
Results are shown in Fig. 8. The methods **I**, **II** and **III** are represented by circle, cross and square symbols, respectively. Similarly to the 1D study (see §4.3.1), method **III** provides  $\mathcal{L}^2$  norm errors reduced by several orders of magnitude compared to method **I**. In the present case, the accuracy of  $\mathcal{O}(10^{-12})$  is reached for values of  $\xi$  of  $\mathcal{O}(10^{-8})$ . However, contrary to the 1D study, the method **II** without iterations provides a better error compared to method **I** for  $\xi > 0.01$ . Of course, the required value of  $\xi$  to reach machine precision accuracy depends on the configuration simulated and the numerical set-up. However it is emphasised that this value lies in the range of  $\mathcal{O}(10^{-8})$  to  $\mathcal{O}(10^{-6})$ .

#### 4.4. Investigation of the RKC method

The damping parameter  $\epsilon$  in Eq. (48) can be freely set by the user, but is critical to control the stability region of the method. As explained by Verwer et al. [38], if  $\epsilon$  is very small, the stability region is extended along the negative real axis of the stability map but has a narrow strip with a sinusoidal shape. On the other hand, if  $\epsilon$  is increased, the stability region becomes wider with an oval shape, but is shortened along the negative real axis. According to Verwer et al. [38], the choice of  $\epsilon$  is crucial because the stability region must include the eigenvalues of both convection and diffusion; smaller values result in large imaginary parts whereas larger values result in large negative real part. Verwer et al. [38] recommend the following choices, depending on the kind of problem solved:  $\epsilon = 2/13$  for strongly diffusion-dominated problems,  $\epsilon = 5$  for convection-dominated problems, and  $\epsilon = 10$  for mixed convection–diffusion problems. The attraction of the RKC method is that it has a stability interval that increases, respect to the number of iterations, with a quadratic behaviour. For very



**Fig. 8.**  $L^2$  norm errors of the velocity field at  $t = 1$  ms formed by methods I, II and III respect to the tolerance parameter  $\xi$  for the 2D vortex-flame test case.



**Fig. 9.** Minimum number of iterations of the RKC method to ensure stability for different values of  $\Delta t$  ranging from  $6.25 \times 10^{-8}$  s to  $2.75 \times 10^{-6}$  s. The configuration investigated is the freely propagating 1D premixed methane flame in a closed, periodic domain (see §4.1.2). Circle symbol (○): number of stages  $K$ . Cross symbol (×): number of stages  $L$ .

large values of  $\epsilon$ , this quadratic behaviour is lost. However, for  $\epsilon = 10$  the stability boundary  $\beta(K)$  follows the relation  $\beta(K) \approx 0.34(K^2 - 1)$ . This means that even for a convection–diffusion problem, the quadratic behaviour of the RKC method is ensured.

As the algorithm developed in the present paper employs the RKC method to solve problems involving convection and diffusion operators of equivalent importance, a value of  $\epsilon = 10$  is chosen. Hence, the stability depends on the number of stages  $K$  and  $L$  chosen for the RKC integration procedures. It is interesting to study the evolution of the minimum number of stages  $K$  and  $L$  required to ensure stability. The freely propagating 1D premixed methane/air flame in a periodic closed domain, detailed above in §4.1.2, is selected as a test case. Recall that  $\Delta x = 20 \mu\text{m}$ . Fig. 9 presents the results for different values of  $\Delta t$  ranging from  $6.25 \times 10^{-8}$  s to  $2.75 \times 10^{-6}$  s. Note that this latter value corresponds to a convective CFL of approximately 0.85 and represents an upper-bound limit on the overall stability of the algorithm. Beyond this value the algorithm is found unstable whatever the choice of  $K$  and  $L$ .

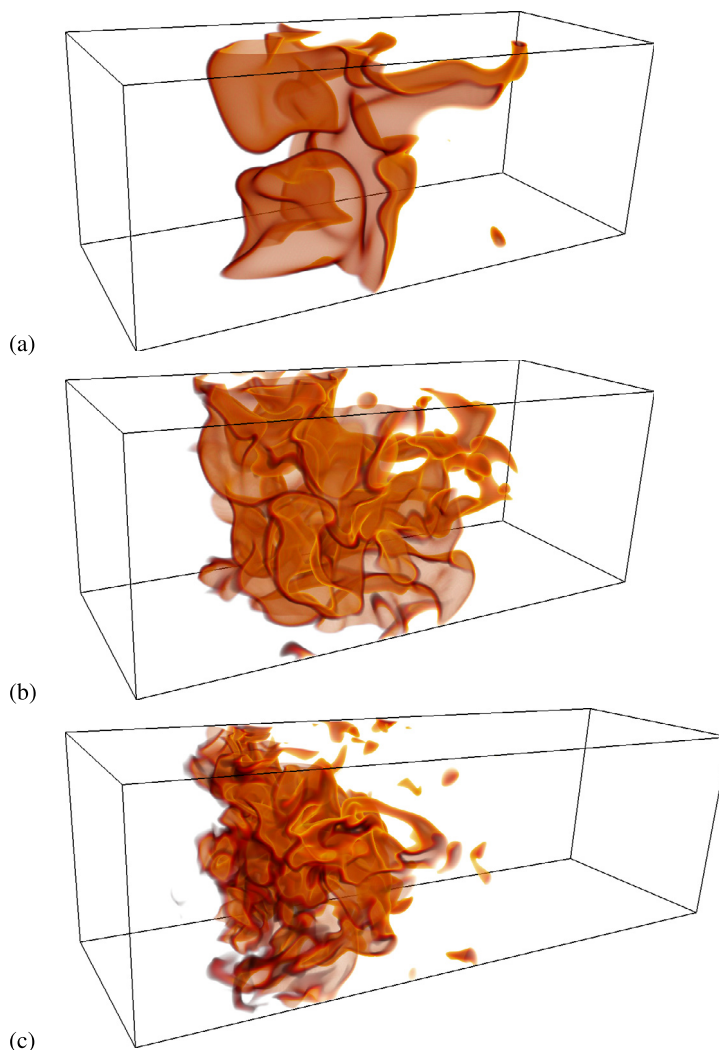
A simple calculation based on the von Neumann stability criterion shows that a maximum time-step of approximately  $2.75 \times 10^{-8}$  s would be required to ensure stability. Recall that the minimum number of stages required for the RKC method is  $K = L = 2$ . This minimum value ensures stability for time-steps below  $\Delta t = 6.25 \times 10^{-8}$  s, which is the limit estimated from the von Neumann stability criterion. As depicted in Fig. 9, employing larger time-steps requires the number of stages of the RKC method to be increased. For a large time-step  $\Delta t = 2.75 \times 10^{-6}$  s,  $K = 14$  and  $L = 12$  are needed to keep the stability of the algorithm.

The present analysis demonstrates the power of the Runge–Kutta–Chebyshev method to perform the integration of the diffusion terms. Only a few iterations are needed to ensure the stability of the algorithm, and for a range of time-steps up to about four orders larger than the critical time-step estimated by the von Neumann stability criterion. Note that although direct comparisons cannot be performed because the test cases are not identical, the present paper shows that the number of iterations required by the RKC method is about an order less than the number of iterations reported by Yu et al. [28] who adopted a fractional step approach to integrate the diffusion operators.

#### 4.5. 3D premixed turbulent flame in a lean methane/air mixture

**HOLMAC** has been developed primarily for applications to DNS of low-Mach-number turbulent reacting flows with multistep kinetics. Fig. 10 shows snapshots from three simulations of a premixed turbulent flame propagating in a three-dimensional domain in a lean methane/air mixture at an equivalence ratio of 0.6. Volume rendering of temperature is shown for three levels of turbulence intensity  $u'$ . When  $u'$  is normalised by the laminar flame speed  $S_L$ , the corresponding values are 2, 5, and 10 in Figs. 10(a), (b), and (c), respectively. The domain is  $12 \times 5 \times 5$  mm in size and uniform discretisation of  $10 \mu\text{m}$  is employed. The flame thickness is of the order of about  $100 \mu\text{m}$  and the Kolmogorov length scale for the highest





**Fig. 10.** Volume rendering of temperature for 3D simulations of premixed turbulent flames in a lean methane/air mixture for normalised turbulence intensities  $u'/S_L$  of (a): 2, (b): 5 and (c): 10.

turbulence intensity case is approximately  $20\ \mu\text{m}$ . In other words, the Kolmogorov length scale and the flame are resolved. The figures are taken once the turbulent flame speed has reached a statistically steady state value. Detailed analysis of these and other results will be published separately. The objective of reporting these results here is only to demonstrate the capability of the code to capture these complex turbulent flame structures. In particular, notice that as turbulence intensity is increased, there is increased wrinkling of the flame. This would, of course, result in increased turbulent flame speed. Also, notice that as turbulence intensity is increased, *islands* of burned and unburned regions can be clearly identified. These islands are formed as a result of flame extinction caused by turbulent strain. Subsequent to extinction, the mixture may locally reignite. Accurate prediction of extinction, ignition and re-ignition often require multistep kinetics. It is interesting to note that the highest absolute turbulence intensity values in these simulations are about  $2\ \text{m s}^{-1}$  and the highest turbulent flame speed is also about  $2\ \text{m s}^{-1}$ . These speeds are sufficiently low for this problem to lie in the low-Mach-number regime. **HOLOMAC** is well suited for these fundamental studies carried out at low-Mach-number and requiring multistep kinetics for accurate prediction of complex phenomena.

## 5. Conclusions

A novel and efficient algorithm with quasi-spectral accuracy has been presented in this paper to conduct DNS of turbulent reacting flows with detailed chemistry under the low-Mach-number assumption. The accuracy and efficiency of the algorithm have been assessed by employing different test problems. First, a freely propagating methane/air premixed flame was computed with a detailed kinetic mechanism. Comparisons have been made with solutions from the reference commercial software **CHEMKIN**, as well as self-convergence tests. It has been demonstrated that a second-order accuracy is reached

in time, which was expected, and this validates the implementation of an operator-split strategy. As the spatial discretisation is performed with high-order compact schemes, it has been proven that the spatial convergence rate is sixth-order when the domain is periodic, and between fourth- and fifth-order accurate when walls or inflow/outflow boundary conditions are employed. A 2D case consisting on the interaction between vortices and a methane/air flame has been performed, demonstrating the ability of the algorithm to handle vortical flow fields in the domain and at boundaries.

In the context of a fractional-step, projection-correction procedure, three different methods have been proposed to resolve the variable-coefficient Poisson equation and to perform the correction in pressure of the velocity fields. An enhanced method, based on a mixed implicit–explicit approach, has been proposed to speed-up the achievement of machine precision accuracy. Furthermore, the convection–diffusion operators are treated with an explicit Runge–Kutta–Chebyshev method. The performance of the method has been assessed and depending on the time-step, only a few number of iterations are required to ensure numerical stability for large time-steps. Finally, 3D Direct Numerical Simulations of a premixed turbulent flame in a lean methane/air mixture and for different turbulence level intensities are reported.

The code developed to implement the present algorithm has been called **HOLOMAC**, and its efficiency opens the way to tackle DNS of reacting flows to understand complex turbulent and chemical phenomena in flames.

## Acknowledgement

The authors wish to thank Prof. Sylvain Laizet (Imperial College of London, UK) for his valuable help about the **Incompact3D** framework and the **2DECOMP&FFT** libraries, Prof. Vinicio Magi (University of Basilicata, Italy) for his comments through the development of the present algorithm, Prof. Lawrence F. Shampine (Southern Methodist University, USA) for his help on the implementation of the RKC method, as well as Dr. Muhsin Mameen and Zhiyan Wang (Purdue University, USA) for providing validation test cases. The authors would also like to thank eResearch South-Australia (eRSA) for providing the computational facilities and their help on technical issues.

## References

- [1] N. Li, S. Laizet, 2DECOMP&FFT – a highly scalable 2d decomposition library and FFT interface, in: Cray User Group 2010 Conference, pp. 1–13.
- [2] H. Pitsch, Large eddy simulation of turbulent combustion, *Annu. Rev. Fluid Mech.* 38 (2006) 453–482.
- [3] L.Y.M. Gicquel, G. Staffelbach, T. Poinso, Large eddy simulations of gaseous flames in gas turbine combustion chambers, *Prog. Energy Combust. Sci.* 38 (2012) 782–817.
- [4] E. Motheau, F. Nicoud, T. Poinso, Mixed acoustic-entropy combustion instabilities in gas turbines, *J. Fluid Mech.* 749 (2014) 542–576.
- [5] B. Franzelli, E. Riber, M. Sanjosé, T. Poinso, A two-step chemical scheme for large-eddy simulation of kerosene–air flames, *Combust. Flame* 157 (2010) 1364–1373.
- [6] S. Mukhopadhyay, J. Abraham, Influence of heat release and turbulence on scalar dissipation rate in autoigniting *n*-heptane/air mixtures, *Combust. Flame* 159 (2012) 2883–2895.
- [7] J. de Charentenay, D. Thévenin, B. Zamuner, Comparison of direct numerical simulations of turbulent flames using compressible or low-Mach number formulations, *Int. J. Numer. Methods Fluids* 39 (2002) 497–515.
- [8] G. Volpe, Performance of compressible flow codes at low Mach numbers, *AIAA J.* 31 (1993) 49–56.
- [9] E. Turkel, Preconditioned methods for solving the incompressible and low speed compressible equations, *J. Comput. Phys.* 72 (1987) 277–298.
- [10] S. Venkateswaran, C. Merkle, Dual time stepping and preconditioning for unsteady computations, *AIAA paper*, 1995, 078.
- [11] J.-S. Shuen, K.-H. Chen, Y. Choi, A coupled implicit method for chemical non-equilibrium flows at all speeds, *J. Comput. Phys.* 106 (1993) 306–318.
- [12] D. Lee, The design of local Navier–Stokes preconditioning for compressible flow, *J. Comput. Phys.* 144 (1998) 460–483.
- [13] A.J. Chorin, A numerical method for solving incompressible viscous flow problems, *J. Comput. Phys.* 135 (1997) 118–125.
- [14] F. Nicoud, Conservative high-order finite difference schemes for low-Mach number flows, *J. Comput. Phys.* 158 (2000) 71–97.
- [15] R. Knikker, A comparative study of high-order variable-property segregated algorithms for unsteady low Mach number flows, *Int. J. Numer. Methods Fluids* 66 (2011) 403–427.
- [16] D. van der Heul, C. Vuik, P. Wesseling, A conservative pressure-correction method for flow at all speeds, *Comput. Fluids* 32 (2003) 1113–1132.
- [17] K. Nerinckx, J. Vierendeels, E. Dick, Mach-uniformity through the coupled pressure and temperature correction algorithm, *J. Comput. Phys.* 206 (2005) 597–623.
- [18] F. Cordier, P. Degond, A. Kumbaro, An asymptotic-preserving all-speed scheme for the Euler and Navier–Stokes equations, *J. Comput. Phys.* 231 (2012) 5685–5704.
- [19] J. Doom, Y. Hou, K. Mahesh, A numerical method for DNS/LES of turbulent reacting flows, *J. Comput. Phys.* 226 (2007) 1136–1151.
- [20] E.A. Sewall, D.K. Tafti, A time-accurate variable property algorithm for calculating flows with large temperature variations, *Comput. Fluids* 37 (2008) 51–63.
- [21] A. Majda, J. Sethian, The derivation and numerical solution of the equations for zero Mach number combustion, *Combust. Sci. Technol.* 42 (1985) 185–205.
- [22] P.A. McMurtry, W.-H. Jou, J. Riley, R.W. Metcalfe, Direct numerical simulations of a reacting mixing layer with chemical heat release, *AIAA J.* 24 (1986) 962–970.
- [23] A.W. Cook, J.J. Riley, Direct numerical simulation of a turbulent reactive plume on a parallel computer, *J. Comput. Phys.* 129 (1996) 263–283.
- [24] H.N. Najm, P.S. Wyckoff, O.M. Knio, A semi-implicit numerical scheme for reacting flow: I. Stiff chemistry, *J. Comput. Phys.* 143 (1998) 381–402.
- [25] O.M. Knio, H.N. Najm, P.S. Wyckoff, A semi-implicit numerical scheme for reacting flow: II. Stiff, operator-split formulation, *J. Comput. Phys.* 154 (1999) 428–467.
- [26] M.S. Day, J.B. Bell, Numerical simulation of laminar reacting flows with complex chemistry, *Combust. Theory Model.* 4 (2000) 535–556.
- [27] C. Safta, J. Ray, H.N. Najm, A high-order low-Mach number AMR construction for chemically reacting flows, *J. Comput. Phys.* 229 (2010) 9299–9322.
- [28] R. Yu, J. Yu, X.-S. Bai, An improved high-order scheme for DNS of low Mach number turbulent reacting flows based on stiff chemistry solver, *J. Comput. Phys.* 231 (2012) 5504–5521.
- [29] V. Giovangigli, *Multicomponent Flow Modeling, Modeling and Simulation in Science, Engineering and Technology*, Birkhauser, Boston, 1999.
- [30] T. Poinso, D. Veynante, *Theoretical and Numerical Combustion*, third edition, 2011, [www.cerfacs.fr/elearning](http://www.cerfacs.fr/elearning).
- [31] A. Ern, V. Giovangigli, Fast and accurate multicomponent transport property evaluation, *J. Comput. Phys.* 120 (1995) 105–116.

- [32] T.E. Magin, G. Degrez, Transport algorithms for partially ionized and unmagnetized plasmas, *J. Comput. Phys.* 198 (2004) 424–449.
- [33] J.O. Hirschfelder, C.F. Curtiss, R.B. Bird, *Molecular Theory of Gases and Liquids*, John Wiley & Sons, New York, 1969.
- [34] E. Hairer, G. Wanner, *Solving Ordinary Differential Equations II - Stiff and Differential-Algebraic Problems*, 2nd edition, vol. 14, Springer-Verlag, Berlin, 1996.
- [35] G. Strang, On the construction and comparison of difference schemes, *SIAM J. Numer. Anal.* 5 (1968) 506–517.
- [36] M. Duarte, S. Descombes, C. Tenaud, S. Candel, M. Massot, Time-space adaptive numerical methods for the simulation of combustion fronts, *Combust. Flame* 160 (2013) 1083–1101.
- [37] H.N. Najm, O.M. Knio, Modeling low Mach number reacting flow with detailed chemistry and transport, *J. Sci. Comput.* 25 (2005) 263–287.
- [38] J.G. Verwer, B.P. Sommeijer, W. Hundsdorfer, RKC time-stepping for advection–diffusion–reaction problems, *J. Comput. Phys.* 201 (2004) 61–79.
- [39] R.J. Kee, F. Rupley, E. Meeks, CHEMKIN-III: a FORTRAN chemical kinetics package for the analysis of gas-phase chemical and plasma kinetics, Technical report SAND96-8216, Sandia National Laboratories, 1996.
- [40] O. Desjardins, G. Blanquart, G. Balarac, H. Pitsch, High order conservative finite difference scheme for variable density low Mach number turbulent flows, *J. Comput. Phys.* 227 (2008) 7125–7159.
- [41] S. Laizet, E. Lamballais, High-order compact schemes for incompressible flows: a simple and efficient method with quasi-spectral accuracy, *J. Comput. Phys.* 228 (2009) 5989–6015.
- [42] S. Laizet, N. Li, Incompact3d: a powerful tool to tackle turbulence problems with up to  $O(10^5)$  computational cores, *Int. J. Numer. Methods Fluids* 67 (2011) 1735–1757.
- [43] M. Frigo, S. Johnson, The design and implementation of FFTW3, *Proc. IEEE* 93 (2005) 216–231.
- [44] P.N. Brown, G.D. Byrne, A.C. Hindmarsh, Vode: a variable-coefficient ode solver, *SIAM J. Sci. Stat. Comput.* 10 (1989) 1038–1051.
- [45] I. Orlanski, A simple boundary condition for unbounded hyperbolic flows, *J. Comput. Phys.* 21 (1976) 251–269.
- [46] J.L. Guermond, P. Mineev, J. Shen, An overview of projection methods for incompressible flows, *Comput. Methods Appl. Mech. Eng.* 195 (2006) 6011–6045.
- [47] P.M. Gresho, Incompressible fluid dynamics: some fundamental formulation issues, *Annu. Rev. Fluid Mech.* 23 (1991) 413–453.
- [48] F.C. Nicoud, Numerical study of a channel flow with variable properties, in: *Annual Research Briefs*, Center for Turbulence Research, NASA Ames/Stanford Univ., 1998, pp. 289–310.
- [49] M.S. Dodd, A. Ferrante, A fast pressure-correction method for incompressible two-fluid flows, *J. Comput. Phys.* 273 (2014) 416–434.
- [50] S. Lele, Compact finite difference schemes with spectral like resolution, *J. Comput. Phys.* 103 (1992) 16–42.
- [51] D.V. Gaitonde, M.R. Visbal, High-order schemes for Navier–Stokes equations: algorithm and implementation into FDL3DI, Technical report, DTIC Document, 1998.
- [52] R. Sankaran, E. Hawkes, J. Chen, T. Lu, C.K. Law, Structure of a spatially developing turbulent lean methane–air Bunsen flame, *Proc. Combust. Inst.* 31 (2007) 1291–1298.
- [53] O. Colin, F. Ducros, D. Veynante, T. Poinot, A thickened flame model for large eddy simulations of turbulent premixed combustion, *Phys. Fluids* 12 (2000) 1843–1863.
- [54] B. Lessani, M.V. Papalexandris, Time-accurate calculation of variable density flows with strong temperature gradients and combustion, *J. Comput. Phys.* 212 (2006) 218–246.
- [55] S. Dong, J. Shen, A pressure correction scheme for generalized form of energy-stable open boundary conditions for incompressible flows, *J. Comput. Phys.* 291 (2015) 254–278.
- [56] W.H. Raymond, H.L. Kuo, A radiation boundary condition for multi-dimensional flows, *Q. J. R. Meteorol. Soc.* 110 (1984) 535–551.
- [57] M. Billson, L.-E. Eriksson, L. Davidson, Acoustic source terms for the linearized Euler equations in conservative form, *AIAA J.* 43 (2005) 752–759.
- [58] N. Peters, G. Paczko, R. Seiser, K. Seshadri, Temperature cross-over and non-thermal runaway at two-stage ignition of *n*-heptane, *Combust. Flame* 128 (2002) 38–59.